HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Electrical and Communications Engineering

Jarno Vanhatalo

Sparse log Gaussian process in Spatial Epidemiology

In partial fulfillment of the requirements for the degree of Master of Science, Espoo August 8, 2006.

Supervisor: Jouko Lampinen

Instructor: Aki Vehtari

TEKNILLINEN KORKEAKOULU **DIPLOMITYÖN TIIVISTELMÄ**

Tekijä: Jarno Vanhatalo

Otsikko: Harva Gaussinen prosessi spatiaalisessa epidemiologiassa

Päivämäärä: 8. elokuuta 2006 **Sivumäärä:** 91

Osasto: Sähkö- ja tietoliikennetekniikan osasto
Professuuri: S-114, Laskennallinen tekniikka

Työn valvoja: Prof. Jouko Lampinen
Työn ohjaaja: TkT Aki Vehtari

Tässä diplomityössä esitetään hierarkinen Bayesilainen malli tautikartoituksen avuksi. Tautikartoitus on spatiaalisen epidemiologian osa-alue, jonka tavoitteena on tutkia terveysriskin maantieteellistä vaihtelua. Tavoitteena on kuvata taudin jakautumista kartalla ja korostaa alueita, joissa tauti- tai kuolemanriski ovat kohonneita.

Tässä työssä käytetään kolmen hierarkiakerroksen mallia tutkimaan kuolleisuusriskin alueellisia vaihteluja kuolleisuusdatasta. Kuolleisuus tietyllä alueella mallinnetaan Poissonin prosessilla, jonka odotusarvo saadaan vakioidun kuolleisuusriskin ja suhteellisen riskin tulona. Kuolleisuusriski vakioidaan taustapopulaation ikä-, sukupuoli- ja koulutustasojakauman avulla. Suhteellisen riskin logaritmille annetaan prioriksi Gaussinen prosessi, joka tasoittaa riskipintaa ja lisää alueiden väliset korrelaatiot malliin. Gaussisen prosessin ongelmaksi muodostuu kovarianssimatriisin inversioon tarvittava aika, jota pienennetään tekemällä Gaussiselle prosessille harva aproksimaatio.

Spatiaalisessa epidemiologiassa on tärkeää pystyä määrittämään tautiriskin alueellisen vaihtelun tilastollinen merkittävyys. Jotta mallin epävarmuusestimaateille saataisiin mahdollisimman hyvät arviot suoritetaan mallin parametrien ylitse integrointi Markov ketju Monte Carlo menetelmiä käyttäen. Gaussisen prosessin latenttien muuttujien näytteistämistä nopeutetaan muunnoksella, joka käyttää hyväkseen posteriorijakauman kovarianssin aproksimaatiota. Markov-ketju-näytteistäminen suoritetaan hybrid Monte Carlo -menetelmällä, jonka oleellinen osa on marginaaliuskottavuuden logaritmin gradienttien laskenta. Harvan aproksimaation tapauksessa gradientit lasketaan muodostamatta eksplisiittisesti täyttä kovarianssimatriisia. Työ esittelee latenttien muuttujien muunnoksen ja gradienttien laskennan toteutukset.

Täyttä ja harvaa Gaussista prosessia käyttäviä malleja testataan kahteen kuolemansyydataan neljällä eri kovarianssifunktiolla, ja malleja verrataan keskenään käyttäen DIC-informaatiokriteeriä. Kuolemansyydatan analyysin tulokset esitetään kuolemanriskikarttoina.

Avainsanat: Alueellinen epidemiologia, tautikartoitus, harva Gaussinen prosessi, Bayesilainen päättely

HELSINKI UNIVERSITY OF TECHNOLOGY

ABSTRACT OF THE MASTER'S THESIS

Author: Jarno Vanhatalo

Title: Sparse Log Gaussian Process in Spatial Epidemiology

Date: August 8, 2006 **Number of pages:** 91

Department: Department of Electrical and Communications Engineering
Professorship: S-114, Computational Engineering

Supervisor: Prof. Jouko Lampinen
Instructor: Dr.Tech. Aki Vehtari

This thesis presents a hierarchical Bayesian model for disease mapping methodology. Disease mapping studies comprise spatial epidemiological methods to summarize the spatial variations in the incidence rate of diseases. The aim is to describe the overall disease distribution on a map and highlight areas of elevated or lowered mortality or morbidity risk.

In this work, a three level hierarchical model is build to study the spatial variations in the relative mortality risk in an areally referenced health-care data. The mortality in an area is modeled as a Poisson process with mean intensity surface, which is a product of a standardized expected number of deaths and a relative risk. The expected number of deaths is evaluated using an age, gender and scholarly degree standardization. The logartihm of the relative risk is given a Gaussian process prior, which smoothes the risk surface and includes the spatial correlation between areas in the model. A problem in Gaussian processes is the computational burden of the required covariance matrix inversion. To overcome the computational problem a fully independent conditional sparse approximation is used.

In spatial epidemiology it is very important to have good estimates whether the spatial variation is significant. To set a golden standard for the uncertainty estimates, both the hyperparameters and the latent values of Gaussian process are marginalized out using Markov chain Monte Carlo methods. The sampling of the latent values is sped up with transformations taking into account the approximate conditional posterior covariance. The sampling is conducted using hybrid Monte Carlo methods which require the gradients of the logarithm of marginal likelihood. The gradients of the sparse approximation are evaluated without forming the full covariance matrix. The work presents an implementation of the gradients and the transformation of latent values for the sparse approximation.

The full and sparse Gaussian models, with four different covariance functions, are applied for two mortality data sets. The models are compared to each others with deviance information criterion and the results of the analysis are presented with maps revealing the relative risk.

Keywords: Spatial epidemiology, disease mapping, sparse Gaussian process, Bayesian inference

Foreword

This masters thesis was carried out in the Laboratory of Computational Engineering at the Helsinki University of Technology, as a part of the New Analysis Methods for Healthcare Process Management (TERENA) project. The research project is a part of the FinnWell Healthcare technology programme, funded by Finnish Funding Agency for Technology and innovation (TEKES).

The mortality data used in the study was acquired from the Statistics Finland. The maps presented in the work are to illustrate the methodological results and they have not been analyzed by healthcare specialists.

I wish to express my sincere gratitude to my instructor, Dr. Aki Vehtari, for his excellent guidance, and support during the work. My supervisor, Prof. Jouko Lampinen, also deserves big thanks for the opportunity to carry out this work and participate the motivating research group.

Special thanks are committed to Harri Valpola for his help in the specific problem of latent value transformation and Markus Siivola for his help and advices on the data manipulation. Finally, I would like to thank all the personnel of the laboratory for nice and welcoming working atmosphere, as well as Sari for the support at home.

In Espoo, August 8, 2006

Jarno Vanhatalo

Contents

1 Introduction **1**

2 Spatial epidemiology **4**

2.1 Focus for the study 4

2.2 Defining spatial data 5

2.3 Health data 6

2.4 Visualizing the data 7

2.5 Description of data used in the study 8

3 Bayesian approach to disease mapping **9**

3.1 Bayesian inference 10

3.1.1 Bayesian approach 10

3.1.2 Posterior analysis and prediction 11

3.1.3 Integrating over the parameters 12

3.1.4 Model comparison 13

3.2 Disease mapping 16

3.2.1 The study focus 16

3.2.2 Earlier works 16

3.2.3 About prior assumptions 17

3.2.4 Generic hierarchical three level model 18

3.2.5 Age adjusted expected value of deaths 19

4 Gaussian processes **21**

4.1 Definition 22

4.2 Full Gaussian process 23

4.2.1 Gaussian processes with normal likelihood . . . 23
4.2.2 Gaussian processes with an arbitrary likelihood . . . 26
4.3 Covariance functions . . . 28
4.3.1 General definitions and characteristics . . . 28
4.3.2 Squared exponential covariance function . . . 29
4.3.3 Exponential covariance function . . . 29
4.3.4 Mátern class of covariance functions . . . 30
4.4 Sparse Gaussian processes . . . 31
4.4.1 About sparse approximations . . . 31
4.4.2 Fully independent training conditional . . . 32
4.4.3 On the choice of the inducing inputs . . . 35

5 Constructing the model **38**
5.1 Data sets studied . . . 38
5.2 Sparse log Gaussian process model . . . 39
5.3 Prior for covariance function parameters . . . 40
5.4 Inducing inputs . . . 40
5.5 Model criticism . . . 42

6 Computational methods **44**
6.1 Implementation issues . . . 45
6.1.1 Implementation environment . . . 45
6.1.2 About computations with matrices and vectors . . . 46
6.2 Markov chain Monte Carlo methods . . . 47
6.2.1 Metropolis Hastings algorithm . . . 49
6.2.2 Gibbs sampling . . . 50
6.2.3 Hybrid Monte Carlo . . . 51
6.2.4 Monitoring convergence . . . 54
6.3 Transformation of latent values . . . 56
6.3.1 Transformation of variables . . . 56
6.3.2 Non-isotropic distribution . . . 57
6.3.3 Approximate posterior variance . . . 58

6.3.4 Transformation in FITC 59
6.4 Gradients of an energy function 63
6.4.1 Gradients with respect to hyperparameters 63
6.4.2 Gradients with respect to latent values 68

7 Results on case problems **69**
7.1 Examples of maps 69
7.2 Sampling from the posterior 73
7.3 Time needed for the sampling 79
7.4 Model comparison 80

8 Conclusions and future work **85**

List of Figures

4.1 An example of GP regression with full GP 26

4.2 An example of GP repression with FITC sparse approximation 37

5.1 The prior distribution for length scale and magnitude of covariance function 41

5.2 The inducing inputs in the case study problems 41

6.1 An example of correlating non-converged and a converged non-correlating sample chain 55

6.2 Two dimensional normal distributions 58

7.1 The median relative risk surface of the cerebral vascular diseases and the surface of $p(\mu > 1)$ in 20km×20km lattice. 71

7.2 The median relative risk surface of the alcohol related diseases and the surface of $p(\mu > 1)$ in 20km×20km lattice. 72

7.3 The median relative risk surface of the cerebral vascular diseases and the surface of $p(\mu > 1)$ in 10km×10km lattice. 74

7.4 The median relative risk surface of the alcohol related diseases and the the surface of $p(\mu > 1)$ in 10km×10km lattice. 75

7.5 The eigenvalues of prior and approximate posterior covariance matrix in case of study problems 76

7.6 CPU-time for one sample from $p(\theta, \mathbf{f}|D)$ as a function of number of inducing inputs with 915 data points and FITC approximation 80

7.7 Posterior distributions of length scale and magnitude in cerebral vascular diseases data 81
7.8 Posterior distributions of length scale and magnitude in alcohol related diseases data 82

List of Tables

7.1 The autocorrelation times for full and FITC sparse Gaussian process in the case of 20km × 20km lattice 78

7.2 The autocorrelation times for full and FITC sparse Gaussian process in the case of 10km × 10km lattice 78

7.3 The DIC statistics in the case of cerebral vascular diseases data and 20km×20km lattice 83

7.4 The DIC statistics in the case of alcohol related diseases data and 20km×20km lattice 83

7.5 The DIC statistics in the case of cerebral vascular diseases and 10km×10km lattice 84

7.6 The DIC statistics in the case of alcohol related diseases and 10km×10km lattice 84

Abbreviations and notations

A	Scale of half-Student's t distribution
$\text{chol}(\cdot)$	Cholesky decomposition operator
$\text{diag}(\cdot)$	Matrix diagonalization operator
D	Observed data
$D(y, \theta)$	Deviance
$D_{\hat{\theta}}(y)$	Deviance at $\hat{\theta}$
$D_{\text{st}}(y, \theta)$	Standardized deviance
$\hat{D}_{\text{avg}}(y)$	Posterior mean deviance
$E(\cdot)$	Energy function
$\mathbb{E}[\cdot]$	Expected value operator
E_i	Standardized expected number of deaths
f	Function value, training case latent value
$\mathbf{f}$	Vector of latent values
f_i	Latent value related to explanatory variables $\mathbf{x}_i$
$\mathbf{f}_*$	Vector of latent values of test cases
$\tilde{\mathbf{f}}$	Latent value transformed with an approximate posterior precision
$g(\cdot)$	Arbitrary function
$\mathcal{GP}(\cdot, \cdot)$	Gaussian process
$\mathcal{H}$	Hamiltonian function
$J_t(\cdot)$	Proposal distribution of Metropolis Hastings algorithm
k	Momentum variable in hybrid Monte Carlo algorithm
$k(\cdot, \cdot)$	Kernel or covariance function

$k_{\text{exp}}(\cdot, \cdot)$	Exponential covariance function
$k_{\text{sexp}}(\cdot, \cdot)$	Squared exponential covariance function
$k_{\nu=3/2}(\cdot, \cdot)$	Mátern $\nu = 3/2$ covariance function
$k_{\nu=5/2}(\cdot, \cdot)$	Mátern $\nu = 5/2$ covariance function
$K(\cdot)$	Kinetic energy function
$\mathbf{K}_{\text{f,f}}$	Covariance matrix of training cases
$\mathbf{K}_{\text{f},*}$	Covariance matrix between training and test cases
$\mathbf{K}_{*,\text{f}}$	Covariance matrix between test and training cases
$\mathbf{K}_{*,*}$	Covariance matrix of test cases
$\mathbf{K}_{\text{u,u}}$	Covariance matrix of inducing variables
$\mathbf{K}(i, j)$	ij'th element of a covariance matrix
$\log(\cdot)$	Natural logarithm
L	Number of leapfrog steps in hybrid Monte Carlo sampling
$m(\cdot)$	Mean function
m	Number of inducing inputs
n	Number of data points
$N(\cdot, \cdot)$	Normal (Gaussian) distribution function
$p(\cdot)$	Probability density function
p_D	Number of effective parameters
Poisson$(\cdot)$	Poisson Distribution function
Pr$[\cdot]$	Probability
q	Position variable in hybrid Monte Carlo algorithm
$q(\cdot \mid \mathbf{u})$	Approximate inducing conditional
$\mathbf{Q}_{a,b}$	$n \times n$ matrix $\mathbf{K}_{a,u}\mathbf{K}_{u,u}^{-1}\mathbf{K}_{u,b}$ of rank m
t	Time
tr$(\cdot)$	Trace operator
$\mathbf{u}$	Vector of inducing variables
$\mathbf{x}_i$	Vector of explanatory variables
$\mathbf{x}_{i,p}$	p'th component of vector of explanatory variables
$\mathbf{x}_i^{\text{new}}$	Unknown, potentially observed vector of explanatory variables
$\mathbf{x}_u$	Vector of inducing inputs

$\mathbf{X}$	Set of vectors of explanatory variables
y_i	Target variable
Y_i	Observed number of deaths
γ	Second level hyperparameters
δt	Short time change, time-step in hybrid Monte Carlo
ϵ	Additive independent noise
θ	Hyperparameters, parameters of covariance function
$\hat{\theta}_{\text{mean}}$	Posterior mean of hyperparameters
$\hat{\theta}_{\text{median}}$	Posterior median of hyperparameters
$\theta^{(t)}$	Sample of Markov chain at time t
λ_{pers}	Adjustment factor for hybrid Monte Carlo with persistence
Λ	Diagonal $n \times n$ matrix diag $\left[\mathbf{K}_{\text{f,f}} - \mathbf{Q}_{\text{f,f}}\right]$
μ_i	Relative risk
ν	Degrees of freedom of half-Student's t distribution
$\pi(\cdot)$	Generic hyperprior
$\sigma_{\cdot}^2$	Magnitude of variance
Σ	Approximate posterior covariance matrix
Σ_{FITC}	Approximate posterior covariance matrix in FITC approximation
Σ_{l}	Approximate likelihood variance matrix
τ	Autocorrelation time of a Markov chain

DIC	Deviance information criterion
DSR	Directly standardized death rate
FITC	Fully independent training conditional
GP	Gaussian process
HMC	Hybrid Monte Carlo
MCMC	Markov chain Monte Carlo
PSRF	Potential scale reduction factor

Chapter 1

Introduction

The creation of maps is an activity almost as old as the recorded history. The earliest examples of maps can be dated to the ancient civilizations in Mesopotamia and Egypt some 5000 years back in time. The ancient maps typically show important features of physical geography, such as mountains and bodies of water, but aspects of human activity were also mapped. Since the construction of maps is tedious work, cartography has dealt most of its history with so-called general maps, which represent simultaneously several, rather stable, geographical phenomena. Around 1800 begun the development of thematic maps, which display the spatial pattern of a single phenomenon. The maps were often stimulated by available data on the environment or society, such as weather or crime rates.

The spatial epidemiology saw the light of day alongside with the development of thematic maps. The world of 19th century was tormented by infectious diseases, such as yellow fewer in the United States and cholera in Europe, and thus the early works in spatial epidemiology were motivated by the desire to map and study the geographical patterns in disease, and to identify risk factors that may explain those patterns. The early works in the spatial epidemiology initiated disease mapping, a major branch of spatial epidemiological studies, in which also this thesis is placed.

Disease mapping studies aim to summarize the spatial variations in the incidence rate, to

identify areas of high and low disease risk. During the last decades the fast improvement in the computer technology has made the collection, storage and analysis of health data easier than ever, and thus led also to an increased interest in disease mapping. At the same time, the ongoing expansion in the available geo-referenced health data has led to a demand for more sophisticated statistical analysis methods. Bayesian statistical inference provides state of the art methods for analyzing complex real life problems and thus is an attractive manner for the modern-day disease mapping.

A substantial feature of the Bayesian data analysis is the quantification of uncertainties with subjective probabilities. The methodological framework of the Bayesian statistics provides a formal theorem to combine prior knowledge and the information in observed data into posterior knowledge with well defined uncertainties. The inference in Bayesian methods typically leads to complex integrals that are usually estimated numerically with stochastic sampling algorithms. The motivation for this work is originated in the desire to study the applicability of Bayesian approach for disease mapping.

Bayesian models have been implemented in disease mapping already for some time, and thus the particular aim of the work was to study the applicability of Gaussian process and its approximation, sparse Gaussian process, to model risk surfaces. Gaussian processes are an attractive manner to construct intensity surfaces for the purposes of disease mapping, but they face severe problems as the size of data increases. To overcome these limitations a number of approximate methods have been suggested in the literature and the sparse approximation used in this work was published just recently.

The main focus of this work is on methodology research and not in the spatial epidemiology particularly. The work introduces an implementation of a recently proposed sparse Gaussian process into the problem of disease mapping and presents solutions to some of the problems faced in the implementation. The method is tested for two case data sets, the mortality due to cerebral vascular and alcohol related diseases in Finland in 1995-1999. The results obtained from the case problems are promising, but reveal some data dependent problems with the sampling algorithm used.

The structure of this thesis is organized as follows. The discussion starts in chapter 2 with

a brief overview on spatial epidemiology and the issues related to it. Additionally, the data set used for the study is described. Chapter 3 gives an introduction in Bayesian inference and model checking. The chapter discusses also disease mapping, in more detail and introduces a general Bayesian approach for it. In chapter 4 the focus is aimed at Gaussian processes and the theory behind the sparse approximation used in the work is revealed. The specific model constructed in the work and the case data sets are presented in chapter 5. The computational methods play an essential role in the implementation and some of the key aspects involved in it are discussed in chapter 6. These include the used Markov chain Monte Carlo methods, discussion on a helpful parameter transformation and on the implementation in Matlab environment. Chapter 7 presents the results on the case problems and chapter 8 provides a conclusion regarding the work.

Chapter 2

Spatial epidemiology

The recent improvements in availability of geographically indexed health and population data together with advances in computing, geographical information systems and statistical methodology have enabled the investigation of spatial variation in disease risk in more realistic manner than ever before. Spatial epidemiology concerns both, describing and understanding the spatial variation in the disease risk. This chapter gives a brief overview to the issues related to the subject. A more detailed treatment on the wide range of matters related to spatial epidemiology is given, for example, by Elliot et al. (2001).

2.1 Focus for the study

Spatial epidemiology concerns the analysis of the spatial/geographical distribution of the incidence of disease (Lawson, 2001). The simplest form of the subject is the use and interpretation of maps of the locations of disease cases. The associated issues within the spatial epidemiology are the map production and the statistical analysis of the mapped data. By the nature of disease maps, many epidemiological concepts also play an important role in the analysis. The map production concerns not only the collation of geographical information, but the visualization of the information as well. The statistical analysis involves the study and use of the spatial statistical methods for the spatial health care data. In

essence, these two different aspects of the subject have their own impact on the methodology. In this work the focus is in the statistical methods of spatial epidemiology and in the aspects related to them.

In any spatial epidemiological analysis, there will be a study focus, which specifies the nature and style of the methods used. The focus consist of hypotheses about the nature of the spatial distribution of the disease examined and these hypotheses can be categorized in three broad classes of study, *disease mapping*, *ecological analysis* and *disease clustering* (Lawson, 2001).

Disease clustering concerns the analysis of abnormal aggregations of disease. In the simplest form the aim is to assess, if there are clusters of elevated incidence of disease, which can not be explained by the normal variation in incidence given the population distribution. More specific cluster studies may also aim to ascertain the location of a possible cluster.

In the ecological analysis the aim is to analyze the relation between the spatial distribution of disease incidence and measured explanatory variables. The analysis is usually carried out at aggregated spatial level, as for example concerning regional incidence compared to the measured explanatory factors at the same region.

The aim of this thesis falls in the category of disease mapping, which concern the use of models to describe the overall disease distribution on the map. Often the object is simply to smooth the map of disease to uncover the underlying structure from the noisy data. The aim may for example be to highlight areas of elevated or lowered risk or to obtain clues to the disease aetiology.

2.2 Defining spatial data

Spatial data can be classified in a various ways depending on the process creating it and the information it contains. First division can be done by the process defining the locations of the data. In *point process* models the point locations, or co-ordinates, of data are thought

to be a realization of a stochastic process whereas in *point level* and *areal* models the locations of the data are known (Banerjee et al., 2004). A point level and point process data are appointed to continuously varying co-ordinates while areal data refers to a finite sub-region of space, as for example county or country.

The information attached to a certain point or an area might be a simple number of occurrences referring to *count data* or in the case of *point* and *areal referenced data* it may also contain additional covariates. In practice the boundary between point and areal referenced data is not always that clear, since as the area becomes small enough it could be considered appointing to a certain point, and vice verse often, especially in spatial epidemiology, it is hard to imagine data that can be appointed strictly to a certain point and not in a finite region.

The data studied in this work contains the information of background population, death rates for diseases and number of explanatory variables in cells of size 250 by 250 meters at minimum. It is referred as point referenced data since the information is appointed to a certain co-ordinate and the cell size is rather small. As mentioned above it could also be referred to areal data, and this sure will be the case when the data is aggregated into larger cells.

2.3 Health data

Health data for a spatial analysis often arise from several different sources and have seldom been collected for spatial epidemiology in particular. Thus a detailed knowledge of the various sources of the data is vital (Staines and Järup, 2001). Unlike physics, for example, where the measurements are taken under controlled conditions focusing on the study problem, in epidemiology the data is often obtained from governmental registers collected for other purposes than epidemiology. Registries usually record all the members of a certain population during a pre-specified time period and thus leave out the people who for some reason failed to be diagnosed, or who were diagnosed before or after the registry.

Social sciences are notorious for their problematic data and epidemiologists are accustomed to collecting and working with data of limited quality. Even in pre-designed experiments the result of the health status of a person is a subject to human errors in diagnosis, and the comparison of results from a study group consist extra noise due to the heterogeneity of the people in the group. The continuous improvements in the computer technology has expanded the possibilities to data collection, storage and linkage, and thus given rise to overflowing upsurge in information available for spatial epidemiology as well. At the same time the development has made even more essential the critical analysis of the data present.

In contrast to many other countries the qualitative and quantitative properties of registry data in Finland are in general very good. A systematic data collection is based on a unique personal identification number assigned to every citizen since 1960s. The personal identification number can be used to link several data-bases, including a registry of coordinates that define the accurate location of the citizen's living space. This provides a great potential for research purposes in general and for studies of spatial properties of social phenomenon in particular. For readers interested more on the subject, a more detailed discussion of issues concerning health and registry data is given, for example, by Lawson (2001); Staines and Järup (2001).

2.4 Visualizing the data

As spatial epidemiology is interested in exploring the spatial variations in disease risk, it is natural to visualize the results of statistical analysis in a map. A map is defined as a collection of spatially defined objects and it is always an approximation of the true spatial phenomenon chosen by the map-maker. Thus the information in a map is a subjective choice, which should be taken into account when analyzing maps.

Not only is the information presented in the map partial, but the visualization of the information has also great influence on how well it can be adopted. Among others the symbols, colors, resolution and scale of the map concurrent on the interpretation of a

map, and therefore, it is essential to pay attention also on the visualization of the results from a spatial analysis. In this work the focus is not in how to make good maps, but readers interested in the subject are advised to see the treatment of, for example, Lawson (2001), MacEachren et al. (1998), Rytkönen (2004) and Monmonier (2004).

2.5 Description of data used in the study

The data used for the case studies comprised of a lattice data set containing mortality and population data from the year 1970 to the end of 1999. The whole country of Finland is included, spanning an area over 1100km in height and more than 600km in width. The standard population is approximately 5 million people and there are around 200 000 deceased for each five-year period. The data lists every death during 1970-2000 and provides snapshots of the population from census surveys conducted every five years.

The data was aggregated by Statistics Finland from point-referenced data into a lattice formed of 250m $\times$ 250m grid cells. Background population and deaths for each cause of death, covering one month segments, were provided as counts pointed to cells.

The data consisted of six covariates. 1) **Age** of an individual, 2) **sex** of an individual, 3) **cause of death** for a deceased individual, assigned according to 54 coded values, 4) **date of death** for a deceased individual, 5) **co-ordinates** of the lattice cell, within which the individual had a home. 6) **scholarly degree** of the individual assigned according to 3 coded values.

Chapter 3

Bayesian approach to disease mapping

Bayesian data analysis comprises practical methods for making inferences from data using probability models for quantities observed and for quantities wished to learn about. The Bayesian analysis is based on the notion of subjective probability, where all probabilities are measured according to ones prior beliefs and observations of past events. This differs fundamentally from the definition of frequentist statistics, where probability is defined as the number of favorable results in a random test conducted infinite number of times. The cornerstone of Bayesian statistical analysis is the Bayes' theorem, named after Reverend Thomas Bayes (c. 1702-1761), which provides a formal way to combine the prior knowledge of model constructor with the observed data by considering all the parameters of the model and the observable quantities random variables. This is another difference between Bayesian and frequentist approach, which makes it straightforward in Bayesian context to express uncertainties mathematically.

This chapter considers the principles of Bayesian inference in general and its application to specific problem of disease mapping. A more general treatment of Bayesian data analysis is given, for example, by (Gelman et al., 2004).

3.1 Bayesian inference

3.1.1 Bayesian approach

The key principle of Bayesian approach is to construct the posterior probability distribution for the unknown entities in a model given the data sample. To use the model, marginal distributions are constructed for all those entities that we are interested in, that is, the end variables of the study. These can be parameters in parametric models, or predictions in (non-parametric) regression or classification tasks.

Use of the posterior probabilities requires explicit definition of the prior probabilities for the parameters. The posterior probability for the parameters in a model M given data D is, according to Bayes' rule,

$$p(\theta|D, M) = \frac{p(D|\theta, M)p(\theta|M)}{p(D|M)}, \tag{3.1}$$

where $p(D|\theta, M)$ is the likelihood of the parameters θ, $p(\theta|M)$ is the prior probability of θ, and $p(D|M)$ is a normalizing constant, called evidence of the model M. The term M denotes all the hypotheses and assumptions that are made in defining the model, like a choice of covariance function for Gaussian process, specific residual model, covariates included in the model and so on. All the results are conditioned on these assumptions. In this notation the normalization term $p(D|M)$ is directly understandable as the marginal probability of the data, conditioned on M. Integrating over all the parameters, comprise

$$p(D|M) = \int_\theta p(D|\theta, M)p(\theta|M)d\theta. \tag{3.2}$$

When having several models, $p(D|M_l)$ is the marginal likelihood of the model l, which can be used in comprising the posterior probabilities of the models, hence the term evidence of the model.

The prior probability of the parameters $p(\theta|M)$ reflects the subjective prior beliefs and knowledge of the model constructor. Most of the time the prior knowledge is not sufficient

enough to specify a fixed prior distribution and the parameters of the prior, called *hyperparameters*, are also given a prior, a *hyperprior*. The hyperprior, when present, modify the posterior as following

$$p(\theta, \gamma \mid D, M) = \frac{p(D \mid \theta, M)p(\theta \mid \gamma)p(\gamma \mid M)}{p(D \mid M)}, \tag{3.3}$$

where γ represents the hyperparameters of θ, or in other words second level hyperparameters.

3.1.2 Posterior analysis and prediction

The result of Bayesian modeling is the conditional probability distribution of unobserved variables of interest, given the observed data. Consider first a nonlinear unknown function f of which there are noisy observations y for certain inputs $\mathbf{x}$. The posterior distribution of a function value f for an input $\mathbf{x}$ given the training data $D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$, is obtained by integrating the predictions of the model with respect to the posterior distribution of the model

$$p(f \mid \mathbf{x}, D, M) = \int p(f \mid \mathbf{x}, \theta)p(\theta \mid D, M)d\theta, \tag{3.4}$$

where θ denotes all the model parameters and hyperparameters of the prior structure.

The probability model for the measurements, $p(y \mid \mathbf{x}, \theta, \gamma, M)$, contains the chosen approximation functions and residual models. It defines also the likelihood part in the posterior probability term, $p(\theta \mid D, M) \propto p(D \mid \theta)p(\theta \mid M)$. In a regression problem with additive noise ϵ,

$$y = f(\mathbf{x}, \theta) + \epsilon, \tag{3.5}$$

the likelihood is straightforwardly obtained from the noise model of ϵ. In the regression problem the function values f itself are the target and the equation 3.4 gives the posterior probability distribution for them.

A different likelihood is obtained for example in a two class classification problem where

the function values are transformed through a logistic transformation and the probability for a binary valued target y being 1 is

$$p(y = 1 \mid \mathbf{x}, \theta) = \left[1 + \exp(-f(\mathbf{x}, \theta))\right]^{-1}. \tag{3.6}$$

Here the function values play a role of help parameters, which are needed in making the decision between y being 1 or 0, and the interpretation of them alone is not that clear anymore.

3.1.3 Integrating over the parameters

The marginalization usually leads to complex integrals that are possible to solve analytically only in the rare case of so-called conjugate prior for likelihood (Gelman et al., 2004), and thus the literature presents multitude of approaches, how the integrals can be approximated. In point estimate approaches the requirement is to give a single best estimate of parameters without integration and obviously there are several candidates for a point estimate, for example mean, median or mode of the posterior. In the classical maximum likelihood (ML) approach the aim is to find a point estimate for the parameters to maximize the likelihood of a model $p(D \mid \theta, M)$. However, the approach is not Bayesian, since firstly the prior assumptions of the parameters $p(\theta \mid M)$ are left out, and more importantly the inference is based on conditioning the data on parameters and not the parameters on the data. The difference between these two interpretations can be understood considering a frequentist framework, where the single best estimate is a random variable such that, when calculated repeatedly for many new data sets its average will be the estimate. The fundamental assumption in the Bayesian framework, however, is that the inference is based on the specific data available at the moment and the estimate obtained is the single best conditioned on the posterior knowledge. Closest to ML estimate in a Bayesian context is the Maximum a Posterior (MAP) approach, where the point estimate maximizes the posterior probability density $p(\theta|D) \propto p(D \mid \theta)p(\theta)$, or minimizes

the negative log-posterior cost function

$$E = -\log\left(p(D \mid \theta)\right) - \log\left(p(\theta)\right), \tag{3.7}$$

In the rest of the text the above function will be called an *energy* function for the reasons to become apparent during the discussion of hybrid Monte Carlo method in section 6.2.3.

A point estimate does not provide information about the shape of the posterior distribution. To get also an estimate for the shape, for example, a normal approximation (e.g. Gelman et al., 2004) can be centered at the posterior mode. Contrary to point estimates the normal approximation gives also an estimate for the variance and thus for the confidential intervals. To find the single best point there are a variety of optimization algorithms presented in the literature, for example a scaled conjugate gradient algorithm (Bishop, 1995).

In a full Bayesian approach no fixed values are estimated for parameters or hyperparameters, but they are marginalized out. Approximations are then needed for the integrations over the hyperparameters to obtain the posterior of parameters and over the parameters to obtain the predictions of the model (Lampinen and Vehtari, 2001). In this work the inference is conducted in a full Bayesian manner by approximating the integrals with Markov chain Monte Carlo methods to be discussed in section 6.2.

3.1.4 Model comparison

There are always many options in setting up a model for any applied problem (e.g. Gelman et al., 2004). Thus there is also a need to compare the usability of the different models in the problem at hand. There are typically two situations in which models are compared. First, a common approach in model construction is to start with a simple model, check its fit to data and then expand it. The original model is then compared to the expanded, more complex model, in order to judge how much have been gained by expanding the model. This generalizes in the case of comparing a set of nested models and judging how much complexity is necessary to fit the data.

The second scenario of model comparison occurs, when two or more non-nested models are compared. In this case, none of the models generalizes the others and the judgment is given which of the models works best. A better approach still would be to construct a larger model that includes all the original models as special cases, after which predictions could be made by integrating over the models similarly as over the hyperparameters.

Model fit can be summarized numerically by a measure such as mean squared error in regression problems or a fraction of misclassified data points in classification. Measures for the predictive ability of the model can also be constructed, for example, by cross-validation or methods using replicates from posterior predictive distribution. Here the method used for model comparison is the deviance information criterion to be discussed next.

Deviance information criterion

Deviance information criterion (DIC) is a measure of model fit proposed by Spiegelhalter et al. (2002). The measure is based on the *Deviance* discrepancy measure, which is defined as minus two times the log-likelihood

$$D(y, \theta) = -2 \log(p(y|\theta)). \tag{3.8}$$

The deviance has an important role in statistical model comparison because, up to a fixed constant that does not depend on θ, the expected deviance equals two times the Kullback-Leibler information of the model. In the limit of large sample sizes, the model with the highest posterior probability will have the lowest Kullback-Leibler information and thus also the lowest expected deviance (e.g. Spiegelhalter et al., 2002; Gelman et al., 2004). The deviance of a model can thus be thought as the loss in information about the true phenomenon when using the model.

Spiegelhalter et al. (2002) define a standardized deviance

$$D_{\text{st}}(y, \theta) = -2 \log(p(y|\theta)) + 2 \log(s(y)), \tag{3.9}$$

where $s(y)$ is some fully specified standardization term that is a function of the data alone. For members of the exponential family with $\mathbb{E}[Y] = \mu(\theta)$ the deviance $D_{\text{st}}(y, \theta)$ is called a *saturated deviance* obtained by setting $s(y) = p(y|\mu(\theta) = y)$.

The expected deviance can be estimated by a point estimate for the parameters such as the posterior mean $\hat{\theta}_{\text{mean}}$. However, as discussed by Spiegelhalter et al. (2002) it is not strictly necessary to use posterior mean as a point estimate for θ and especially in the case of exponential family likelihood a posterior median $\hat{\theta}_{\text{median}}$ may by justified. The deviance at $\hat{\theta}$ is denoted as

$$D_{\hat{\theta}}(y) = D(y, \hat{\theta}). \tag{3.10}$$

In the Bayesian context, it is appealing to average the deviance itself over the posterior distribution to obtain the *posterior mean deviance*. As will be discussed in the context of Markov chain Monte Carlo methods in the section 6.2, this can be approximated using the posterior simulations of θ

$$\widehat{D}_{\text{avg}}(y) = \frac{1}{N} \sum_{t=1}^{N} D(y, \theta^{(t)}). \tag{3.11}$$

The difference between the posterior mean deviance and the deviance at $\hat{\theta}$ represents the effect of the model fitting and can be used as a measure of the *effective number of parameters*

$$p_D = \widehat{D}_{\text{avg}}(y) - D_{\hat{\theta}}(y). \tag{3.12}$$

The deviance information criterion is defined as the deviance at $\hat{\theta}$, plus twice the effective number of parameters

$$DIC = D_{\hat{\theta}}(y) + 2p_d \tag{3.13}$$

$$= \widehat{D}_{\text{avg}}(y) + p_d, \tag{3.14}$$

where the deviance can be either the classical or standardized deviance defined above. DIC can be considered as a Bayesian measure of fit or adequacy, penalized by an additional complexity term p_D.

3.2 Disease mapping

3.2.1 The study focus

Disease mapping concerns the use of statistical models to describe the overall disease distribution on the map. The study focus is to analyze spatial variations in disease risk, within which different formats of epidemiological data naturally give rise to different statistical methods. Often the object is simply to smooth the map of disease to uncover the underlying structure from the noisy data. The aim may for example be to highlight areas of elevated or lowered risk or to obtain clues to the disease aetiology.

3.2.2 Earlier works

The origins of disease mapping can be traced back to the 19th century, when frequently raging infectious diseases tormented the countries of the time, particularly yellow fewer in the United States and cholera in Europe. One of the most famous early epidemiologists was John Snow (1813-1858), who demonstrated the spread of cholera, through contaminated water in London (Walter, 2001). With his early spot maps of locations with cholera infections Snow was able to show that cholera infections were much more frequent in certain areas of the city. The cause of the increased infection rate was then tracked to the contaminated parts of the Thames.

In the 20th century the research focus shifted from infectious diseases towards chronic diseases such as cancer and heart diseases. In Great Britain, for example, a number of cancer mortality maps were produced for England and Wales in 1920s and 1930s. In early 19th hundreds an important methodological advance was an adjustment for regional differences in age and sex, thus avoiding possibly biased comparison of crude rates of earlier works. By the 1980s some of the first maps with statistical spatial analysis were constructed and the first works with maps showing time trend patterns of diseases were also published. The fast increase in computational power in the late 19 hundreds made the development and use of more sophisticated statistical methods possible. One of the

first empirical Bayesian smoothing techniques was used by Devine *et al.* in 1991 in the United States (Walter, 2001).

In Finland the systematic collection of public health data has been conducted already for several decades. For example E. Pukkala *et al.* in 1987 was able to summarize cancer incidence data since as early as 1953 (Walter, 2001), producing one of the first time trend and widest time ranges covered cancer atlases in Europe. The work by E. Pukkala *et al.* was also one of the earliest to use data smoothing, with a geometric centroid approach. In past few years there have been several public health applications for disease mapping in Finland. For example Bayesian studies of Type I diabetes mellitus (Rytkönen, 2004; Moltchanova, 2005) and studies of acute myocardial infarction in eastern Finland (Karvonen et al., 2002) to mention few of them.

3.2.3 About prior assumptions

The usual assumptions in the model construction are that the measured death rates are a combination of two different (stochastic) processes, the other governing the expectation of mortality and the other the actual death rates in a certain area. The expectation of the mortality is as an intensity surface getting high values in the areas where the prior belief suggests large numbers of death cases. For example in big cities, with a high density of population, there are more death cases than in rural areas with less people. In addition to the background population density the properties of the population might have affect on the intensity as well, and thus these properties could be used as an explanatory covariate. For example age, sex, education and social status of people can be included in such covariates. Also it is possible to add environmental causes such as lakes, industry or main traffic routes in the model. In addition to including explanatory covariates into the model, an important purpose of the intensity surface is to give a ways of smoothing the data and of describing the areal correlations and some stochastic randomness.

The intensity surface represents the number of death cases expected to realize in a certain area. Now, the measured data is not assumed to be explained totally by the intensity

surface, not even the surface was a true one. As in every natural process there are some irregularities present in the data and the stochastic process build for the occurred death cases given the expected value, presents the assumptions about these irregularities. These might include both the noise present in the data and the stochastic randomness of the underlying process.

3.2.4 Generic hierarchical three level model

A widely discussed generic three level hierarchical model for disease mapping based on aggregation of the underlying individual level risk can be summarized as (Best et al., 2005)

$$Y_i \sim \text{Poisson}(E_i \mu_i) \tag{3.15}$$

$$\log(\mu_i) \sim p(\cdot|\theta), \tag{3.16}$$

$$\theta \sim \pi() \tag{3.17}$$

where Y_i is the observed number of deaths, E_i the *standardized expected number of deaths* and μ_i the *relative mortality risk* in an area A_i. The generic prior $p(\cdot|\theta)$ is an appropriate second level prior for log relative risk and θ represents the hyperparameters with prior $\pi()$. The *homogeneous Poisson process*, more commonly Poisson process, is the most commonly used theoretical model for the generation of disease cases (Best et al., 2005). Due to the fact that the population sizes in general are large and the number of disease cases relatively small the Poisson distribution can be considered as a good approximation for the underlying binomial distribution. The standardized expected number of deaths is evaluated from the explanatory *covariates* present in the data and as discussed earlier there is a multitude of ways to construct the standardization. The role of μ is to model the risk relative to E, a deterministic function of explanatory variables.

The difference between most common disease mapping models is in the prior given for $\log(\mu)$. A common second level prior is a *Conditional Autoregressive* (CAR) model, used for example by Richardson et al. (2004) and Moltchanova (2005). The model is based on

evaluating the variance of log relative risk in an area as an average between its neigbours. CAR model is a generic name for a class of models using the same approach for variance evaluation. The model has its main difficulties with a sparse data, where some or many of the neigbours are empty. A *multivariate normal* prior given as $\log(\mu) \sim N(\mathbf{m}, \mathbf{K})$, is one of the most flexible distribution for representing correlated random variables and Gaussian processes used in this work are its extension to a continuous surfaces (Best et al., 2005).

3.2.5 Age adjusted expected value of deaths

In epidemiology most health rates are strongly age-dependent. Most commonly, resulting from the slow deterioration of human biological system, the older age groups generate higher death rates than younger ones. The opposite could take place, for example, in accidental death causes.

The *crude death rate*, the total number of deaths divided by the number of people, is a widely used measure of mortality. However, crude death rates are influenced by the age composition of the population and as such, the comparisons of crude death rates over time or between groups may be misleading if the populations compared differ in age composition. (Anderson and Rosenberg, 1998)

The crude rate is in most cases inadequate to describe mortality risk across the whole population and an age-standardization is a method used to address this problem by defining more detailed rates that better reflect the age composition of the population. Age-standardization is based on death rates that are separately calculated for different age groups. The population is usually apportioned to five year segments, for example below 5, 5-9, 10-14, ..., 80-84 and over 84 years (Anderson and Rosenberg, 1998). The selection of appropriate groupings is not strict but should reflect the average age-structure of the population.

In this work the standardization is done following the idea of the *directly standardized*

rate DSR (Ahmad et al., 2000). The directly standardized death rate for an area A_i is

$$\text{DSR} = \sum_{r=1}^{R} \frac{Y_r}{N_r} \frac{n_{ri}}{\sum_{r=1}^{R} n_{ri}}, \tag{3.18}$$

where Y_r and N_r are the total number of deaths and people in the whole area of study in the age-group r, and n_{ri} is the number of people in the age-group r and in the are i. Now the age adjusted expected value in an area A_i is obtained as the product of the rate and the population in the area,

$$E_i = \sum_{r=1}^{R} \frac{Y_r}{N_r} n_{ir}. \tag{3.19}$$

The standardization in this work uses the information about age, gender and scholarly degree of the people. The population under study was first divided between genders. Both genders were then partitioned into 14 age segments accounting in 28 age-gender groups. All the age-gender groups were still partitioned with respect to 3 scholarly degrees accounting into 66 age groups in total, since not all scholarly degrees are present for all age-gender groups.

Chapter 4

Gaussian processes

Gaussian processes (GP) (e.g. Rasmussen and Williams, 2006) are a flexible and attractive method for a wide variety of supervised learning problems, such as regression and classification in machine learning or spatial analysis in epidemiology. They have been studied already for some time, but due to the fast increase in memory requirements and computational demands as the function of the number of training cases, they have been competitive only in problems with a moderate size dataset. Recently there has been an increasing interest in GPs due to the approximate methods which reduce the computational load.

In this chapter, the Gaussian processes are discussed following the treatment of Quinonero-Candela and Rasmussen (2005) and Rasmussen and Williams (2006). First the definition of Gaussian processes and some fundamental theory behind them are considered. The use of Gaussian processes is discussed with a simple regression problem, after which the consideration is extended into applications with an arbitrary likelihood. The covariance functions used in the work are discussed shortly and at the end of the chapter the focus is taken into sparse approximations and in particularly in the approximation used in this work. The treatment of Rasmussen and Williams (2006) is build up around training GP to find a point estimate for parameters. In this work the aim is, however, on the full Bayesian inference by integrating over the parameters.

4.1 Definition

Whereas a probability distribution describes the properties of random variables, a stochastic process governs the properties of functions. A Gaussian process is a generalization of a Gaussian distribution, which can be understood, for example, by considering a set of explanatory variables, $\mathbf{X} = \{\mathbf{x}_i, ..., \mathbf{x}_n\}$ that are mapped to some function values $f(\mathbf{x}_i)$. A Gaussian process $\mathcal{GP}$ defines the probability of continuous set of function values $f(\mathbf{x}_i)$ indexed by the explanatory variables $\mathbf{X}$, whereas a (one dimensional) Gaussian distribution could be used to define the probability of a one function value $f(\mathbf{x}_i)$ given explanatory variable $\mathbf{x}_i$. Formally a Gaussian process is defined as following (Rasmussen and Williams, 2006):

Definition 1 *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

A Gaussian process is a fully probabilistic model that is completely defined by its mean function, $m(\mathbf{x})$, and covariance function, $k(\mathbf{x}_i, \mathbf{x}_j)$, defined

$$m(\mathbf{x}) = \mathbb{E}\left[f(\mathbf{x})\right] \tag{4.1}$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbb{E}\left[(f(\mathbf{x}_i) - m(\mathbf{x}_i))(f(\mathbf{x}_j) - m(\mathbf{x}_j))\right]. \tag{4.2}$$

In a Bayesian framework GP can be used to define a prior distribution over a set of functions

$$p(f \mid \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n) \sim \mathcal{GP}\left(m(\mathbf{x})), k(\mathbf{x}, \mathbf{x}')\right), \tag{4.3}$$

which map the explanatory variables $\mathbf{x}_i$ into the function values $f_i = f(\mathbf{x}_i)$ of interest. The properties such as smoothness and differentiability of the functions restricted by $\mathcal{GP}$ can be varied with the choice of the covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$. Although GPs are very flexible models they are still limited by the form of the covariance function. For example, it is difficult to model non-stationary processes with GP because it is hard to construct useful non-stationary covariance functions.

The definition of GP as a collection of random variables automatically implies a *consistency* requirement, which is also known as the marginalization property. The consistency property means that if a GP specifies for example $(f_1, f_2) \sim \mathrm{N}(\mathbf{m}, \mathbf{K})$, then it must also specify $f_1 \sim \mathrm{N}(m_1, \mathbf{K}_{1,1})$, where $\mathbf{K}_{1,1}$ is the relevant submatrix of the covariance matrix $\mathbf{K}$ (Rasmussen and Williams, 2006).

4.2 Full Gaussian process

4.2.1 Gaussian processes with normal likelihood

Probably the easiest and most intuitive problem to implement for Gaussian process is a regression problem with additive and independent Gaussian noise. Given a training data $D = (\mathbf{x}_i, y_i), i = 1, ..., n$ of n pairs of explanatory variables (inputs) $\mathbf{x}$ and targets y, a predictive distribution of the function values f_* at test locations $\mathbf{x}_*$ is computed. The function values of test cases, $f_{*,i}$, or training cases, f_i, are also called as *latent values* and they represent the noiseless underlying phenomenon under the noisy targets

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \text{ where } \epsilon_i \sim N(0, \sigma^2_{\text{noise}}), \tag{4.4}$$

where σ^2_{noise} is the variance of the noise. To construct a Bayesian inference first the latent values $\mathbf{f} = \left[f_1, f_2, ..., f_n\right]^T$ are given a Gaussian process prior

$$p(\mathbf{f} \mid \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n) \sim \mathcal{GP}(\mathbf{0}, \mathbf{K}), \tag{4.5}$$

where the entries of the covariance matrix $\mathbf{K}_{ij}$ are given by the covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$. For simplicity the mean of the GP is defined here to be zero, which does not restrict the generality of the treatment but makes the equations easier to follow. At this point the dependence of the covariance function parameters and their hyperparameters is omitted. In contrast to parametric models, such as for example linear regression, in which the prior is defined over the parameter values of a fixed function, the GP restricts the study on certain kind of functions defined by the mean and the covariance functions.

The consistency property implies that the joint prior of the training and test cases can be written as

$$p(\mathbf{f}, \mathbf{f}_*) = \mathcal{GP}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathrm{f,f}} & \mathbf{K}_{\mathrm{f},*} \\ \mathbf{K}_{*,\mathrm{f}} & \mathbf{K}_{*,*} \end{bmatrix}\right), \tag{4.6}$$

where the covariance matrix is partitioned into four submatrices, whose subscript define the variables between which the correlation is computed. For example $\mathbf{K}_{\mathrm{f},*}$ defines the covariance matrix between training and test latent values. Here it should also be noted that due to the symmetry property of covariance matrix $\mathbf{K}_{f,*}^{\mathrm{T}} = \mathbf{K}_{*,\mathrm{f}}$ (see equation (4.2)). The likelihood is defined by the noise model (4.4) to be also Gaussian with mean $\mathbf{f}$

$$p(\mathbf{y}|\,\mathbf{f}) = \mathrm{N}(\mathbf{f}, \sigma_{noise}^2\mathbf{I}), \tag{4.7}$$

where $\mathbf{I}$ is the identity matrix. By combining the prior and the likelihood, the joint posterior of latent values can be obtained using the Bayes rule (3.1)

$$p(\mathbf{f}, \mathbf{f}_*|\mathbf{y}) = \frac{p(\mathbf{y}|\,\mathbf{f})p(\mathbf{f}, \mathbf{f}_*)}{p(\mathbf{y})}, \tag{4.8}$$

where $p(\mathbf{y}) = \int p(\mathbf{y}|\,\mathbf{f})p(\mathbf{f})d\,\mathbf{f}$ is the marginal likelihood from equation (3.2). To complete the Bayesian inference for the desired posterior predictive distribution of test variables, the unwanted training set latent variables are marginalized out

$$p(\mathbf{f}_*\,|\mathbf{y}) = \int p(\mathbf{f}, \mathbf{f}_*|\mathbf{y})d\mathbf{f} = \frac{1}{p(\mathbf{y})}\int p(\mathbf{f}, \mathbf{f}_*)\,p(\mathbf{y}|\,\mathbf{f})d\mathbf{f}, \tag{4.9}$$

which, since both factors in the integral are Gaussian, can be evaluated in the closed form to give the Gaussian posterior predictive distribution

$$p(\mathbf{f}_*\,|\,\mathbf{y}) = N\left(\mathbf{K}_{*,\mathrm{f}}(\mathbf{K}_{\mathrm{f,f}} + \sigma_{\mathrm{noise}}^2\mathbf{I})^{-1}\mathbf{y},\, \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathrm{f}}(\mathbf{K}_{\mathrm{f,f}} + \sigma_{\mathrm{noise}}^2\mathbf{I})^{-1}\,\mathbf{K}_{\mathrm{f},*}\right). \tag{4.10}$$

The predictive distribution of test targets $p(\mathbf{y}_*\,|\,\mathbf{y})$, can be computed easily by adding the noise $\sigma_{\mathrm{noise}}^2\mathbf{I}$ into the variance in the expression of $p(\mathbf{f}_*\,|\mathbf{y})$.

At this point it can also be noticed, why GPs are considered as non-parametric models. As can be seen it is possible to express the prior without any parametric assumptions. This

far, though, the inference presented has been incomplete by leaving out the consideration of the specific form of GP used, given in the form of the covariance function, and giving only a general result. When taking in also the (necessary) covariance function parameters it can be seen that they play a role similar to hyperparameters in parametric models such as for example the hyperparameters of weights in MLP networks. So, the integration over parameters is done in (4.10) and what is left is the inference on hyperparameters. The use of Gaussian process for regression problem is illustrated in the picture 4.1.

To complete the inference, the prior for hyperparameters is included in the model to give a joint posterior

$$p(\mathbf{f}, \mathbf{f}_*, \theta|\mathbf{y}) = \frac{p(\mathbf{y}|\,\mathbf{f})p(\mathbf{f}, \mathbf{f}_* \mid \theta)p(\theta)}{p(\mathbf{y})}, \tag{4.11}$$

and a predictive distribution

$$p(\mathbf{f}_*\,|\mathbf{y}) = \int p(\mathbf{f}, \mathbf{f}_*, \theta|\mathbf{y})d\mathbf{f}d\theta = \frac{1}{p(\mathbf{y})}\int p(\mathbf{y}|\,\mathbf{f})p(\mathbf{f}, \mathbf{f}_* \mid \theta)p(\theta)d\mathbf{f}d\theta, \tag{4.12}$$

where θ represents all the covariance function parameters and hyperparameters. Here the integration over latent values can again be conducted analytically as in (4.10), but the integration over hyperparameters is usually not analytically tractable, which results in various approximations. In GP regression the likelihood times the prior is a product of two Gaussian distributions resulting as well in a Gaussian distribution $p(\mathbf{y}|\,\mathbf{f})p(\mathbf{f}) \sim N(\mathbf{0}, \mathbf{K}_{\mathrm{f,f}} + \sigma^2_{\mathrm{noise}}\mathbf{I})$. Now the energy function (3.7) needed in, for example, Markov chain Monte Carlo methods, is obtained a particularly easy form

$$\begin{aligned} E &= -\log\left(p(\mathbf{y}|\,\mathbf{f})p(\mathbf{f} \mid \theta)\right) - \log\left(p(\theta)\right) \\ &= -\frac{1}{2}\log\left|\mathbf{K}_{\mathrm{f,f}} + \sigma^2_{\mathrm{noise}}\mathbf{I}\right| - \frac{1}{2}\mathbf{y}^{\mathrm{T}}\left(\mathbf{K}_{\mathrm{f,f}} + \sigma^2_{\mathrm{noise}}\mathbf{I}\right)^{-1}\mathbf{y} - \frac{n}{2}\log(2\pi) - \log\left(p(\theta)\right), \end{aligned} \tag{4.13}$$

which is independent of the new cases $\mathbf{f}_*$. The prior hierarchy could be extended also to the level of the hyperparameters of covariance function parameters. However, in this work the prior structure is constructed only to the first level hyperparameters θ. The likelihood term in (4.13) can also be called a marginal likelihood, since it is already marginalized analytically over the latent values. This is the interpretation of, for example, Rasmussen

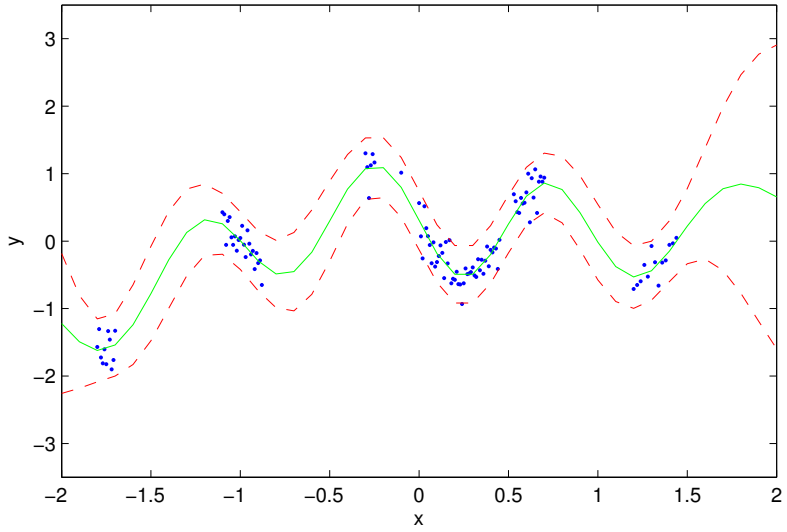


Figure 4.1: **An example of GP regression with full GP**. The data points are marked with blue dots, the *green* line represents the output f of trained GP and dashed red lines are the $f \pm 2\sigma$, where the σ is the standard deviation predicted by the model.

and Williams (2006).

4.2.2 Gaussian processes with an arbitrary likelihood

The inference for GP with an arbitrary likelihood follows closely the steps in that of regression. The main difference is that the latent values can not be thought as noiseless target values any longer. To construct the GP inference, first the target values y_i are again given a probabilistic model that depends on the latent values

$$y_i \sim p\left(g(f_i) \mid \mathbf{x}_i\right), \tag{4.14}$$

which is the likelihood of y_i with a parameter $g(f_i)$, where function $g(\cdot)$ can be any function of latent value f_i associated with input $\mathbf{x}_i$. In the regression problem the noise is modeled to be additive as in (4.4), but here it is included in the model $p\left(g(f_i) \mid \mathbf{x}_i\right)$. The model (4.14) can be for example a logistic transformation (3.6), where g is an identity function.

The next step, as in regression problem, is to give a Gaussian process prior for latent values $p(\mathbf{f} \mid \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n) \sim \mathcal{GP}(\mathbf{0}, \mathbf{K})$. The latent values can be considered now as underlying help parameters, similar to the hyperparameters, from which the name *latent*. Since the target values y are not normally distributed given the latent values the posterior predictive distribution

$$p(\mathbf{f}_* \,|\mathbf{y}) = \int p(\mathbf{f}, \mathbf{f}_*|\mathbf{y}) d\mathbf{f} = \frac{1}{p(\mathbf{y})} \int p(\mathbf{f}, \mathbf{f}_*) p(\mathbf{y}|g(\mathbf{f})) d\mathbf{f}, \tag{4.15}$$

can no longer be solved analytically. In this case the integration over latent values can be conducted by, for example, MCMC methods as in the case of hyperparameters. The energy function is now modified into

$$\begin{aligned} E &= -\log\left(p(\mathbf{y}|g(\mathbf{f}))\right) - \log\left(p(\mathbf{f} \mid \theta)\right) - \log\left(p(\theta)\right) \\ &= -\log\left(p(\mathbf{y} \mid g(\mathbf{f}))\right) - \frac{1}{2}\log\left|\mathbf{K}_{\mathrm{f,f}}\right| - \frac{1}{2}\mathbf{f}^{\mathrm{T}}\,\mathbf{K}_{\mathrm{f,f}}^{-1}\,\mathbf{f} - \frac{n}{2}\log(2\pi) - \log\left(p(\theta)\right), \end{aligned} \tag{4.16}$$

where the minus log likelihood $\log\left(p(\mathbf{y} \mid g(\mathbf{f}))\right)$ is explicitly shown in the equation and when compared to (4.13) the covariance matrix $\mathbf{K}_{\mathrm{f,f}} + \sigma^2_{\mathrm{noise}}\mathbf{I}$ is replaced by the prior covariance $\mathbf{K}_{\mathrm{f,f}}$ and the test cases $\mathbf{y}$ are replaced by the latent values.

A variety of approximative methods other than Markov chain Monte Carlo methods for the integral (4.15) are presented in the literature. Minka (2001) has proposed the iterative Expectation propagation algorithm (EP) in which posterior of latent values is approximated by a product of normal distributions centered at points that are sought with an iterative algorithm. The algorithm is successfully implemented, for example, for Gaussian processes in two class classification problems with probit likelihood. Other variational analytic approximation is the Laplace's Method (e.g. Williams and Barber, 1998) or normal approximation mentioned in section 3.1.3. Here the integration is conducted via Markov chain Monte Carlo sampling to obtain golden standard results for the problem, but EP and Laplace methods could be tested later.

4.3 Covariance functions

4.3.1 General definitions and characteristics

An arbitrary function of inputs $\mathbf{x}_i$ and $\mathbf{x}_j$ will not in general be a valid covariance function. In this section, some of the basic requirements and properties of covariance functions will be discussed, after which the covariance functions used in the work will be considered briefly. For more extensive discussion on the subject, see the treatment of, for example, Rasmussen and Williams (2006) or Abrahamsen (1997).

A general name for a function k of two arguments mapping a pair of inputs $\mathbf{x}_i \in \mathbb{R}^n$, $x_j \in \mathbb{R}^n$ into $\mathbb{R}$ is a *kernel*. A sufficient and necessary condition for a kernel k to be a covariance function of consistent finite-dimensional distribution is the *positive semidefiniteness* of the kernel (e.g Abrahamsen, 1997). A kernel is said to be symmetric if $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_i)$, and clearly, from the definition (4.2), covariance functions are symmetric. If the kernel k is a covariance function and there is a matrix $\mathbf{K}$ whose entries are $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, the matrix is called a covariance matrix.

A covariance function is called *stationary* if it is a function of $\mathbf{x}_i - \mathbf{x}_j$, which is invariant to translations in the input space. Further, the covariance function is *isotropic*, if it is a function only of $|\,\mathbf{x}_i - \mathbf{x}_j\,|$, and thus it is invariant to all rotations in the input space. For example a squared exponential covariance function to be discussed later is both stationary and isotropic. The covariance functions can be combined as new covariance functions. For example the sum or product of two covariance functions can be used to make a new covariance function.

The smoothness properties of the Gaussian process are determined by the properties of the covariance function around $\mathbf{0}$ and they can be summarized with terms of a mean square differentiability of a Gaussian process and a differentiability of a covariance function. The mean square differentiability of a process is a stronger property than the differentiability of a covariance function and it is discussed in more detail, for example, by Rasmussen and Williams (2006). The smoothness of the process then has influence on, how fast varying

effects the process can adapt.

4.3.2 Squared exponential covariance function

Probably the most widely-used covariance function is a squared exponential defined as

$$k_{\text{sexp}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2_{\text{sexp}} \exp\left(-\frac{1}{l^2}\sum_{p=1}^{P}(x_{i,p} - x_{j,p})^2\right), \tag{4.17}$$

where l is the *length scale* and σ^2_{sexp} is the *magnitude*. The length scale governs the distance, how far apart inputs still correlate. The role of magnitude can be understood by considering a covariance function that is sum of two kernels, in which case the magnitude describes how much either of the two parts describe of the whole covariance.

A squared exponential covariance function is infinitely differentiable leading to very smooth Gaussian processes that are infinitely mean square differentiable. The covariance function is stationary and isotropic in its basic form. The squared exponential, as all the other covariance functions discussed here, can be modified into a non-isotropic form by setting a different length scale for all the components of $\mathbf{x}$. This is referred as an *automatic relevance determination* kernel discussed by, for example, Neal (1996). In this work all the covariance matrices are stationary and isotropic.

4.3.3 Exponential covariance function

The exponential covariance function is given as

$$k_{\text{exp}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2_{\text{exp}} \exp\left(-\frac{1}{l}\sqrt{\sum_{p=1}^{P}(x_{i,p} - x_{j,p})^2}\right). \tag{4.18}$$

Even the exponential covariance function is infinitely differentiable likewise the squared exponential, a Gaussian process defined by it is not mean squared differentiable. Thus a Gaussian process with an exponential covariance function is not as smooth as one with

a squared exponential. This means that GP with an exponential covariance function can adapt to a rougher phenomenon than with a squared exponential.

4.3.4 Mátern class of covariance functions

The Mátern class of covariance functions is given by

$$k_{\text{matern}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_{\text{m}}^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{l} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu} r}{l} \right), \tag{4.19}$$

where l is the length scale, ν a positive parameter, $r = |\mathbf{x}_i - \mathbf{x}_j|$ and K_{ν} a modified Bessel function (e.g. Abramowitz and Stegun, 1970). This covariance function has the property that as $\nu \to \infty$ it approaches a squared exponential and as $\nu \to \frac{1}{2}$ it approaches an exponential covariance function. A Gaussian process with Mátern class covariance function is k times mean square differentiable if $\nu > k$. Thus the smoothness properties of the GP with a Mátern covariance function can be controlled with the parameter ν.

The Mátern covariance functions can be computed faster when ν is a half integer $\nu = p + 1/2$, where p is a positive integer (e.g. Rasmussen and Williams, 2006). The general expression can be derived into

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_{\text{m}}^2 \exp\left(-\frac{\sqrt{2\nu} r}{l} \right) \frac{\Gamma(p+1)}{\Gamma(2p+1)} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{\sqrt{8\nu} r}{l} \right)^{p-i} \tag{4.20}$$

The Mátern class covariance functions used in this work have $\nu = 3/2$ and $\nu = 5/2$, and can be represented with the above as

$$k_{\nu=3/2}(y_i, y_j) = \sigma_{\nu=3/2}^2 \left(1 + \frac{\sqrt{3} r}{l} \right) \exp\left(-\frac{\sqrt{3} r}{l} \right) \tag{4.21}$$

$$k_{\nu=5/2}(y_i, y_j) = \sigma_{\nu=5/2}^2 \left(1 + \frac{\sqrt{5} r}{l} + \frac{5r^2}{3l^2} \right) \exp\left(-\frac{\sqrt{5} r}{l} \right) \tag{4.22}$$

4.4 Sparse Gaussian processes

4.4.1 About sparse approximations

The main drawback of a full Gaussian process is the fast growing need of computation time and memory requirements as the size of the training set increases. The memory requirements for storing and the time for inversion of an $n \times n$ covariance matrix, needed for example in (4.16), grow respectively as $O(n^2)$ and $O(n^3)$ with respect to the size of the training data n. Sparse Gaussian processes are a class of approximations in which the full covariance matrix is given a reduced rank approximation in order to speed up the computations.

The simplest possible sparse approximation would be to use only a subset of the training data. In this approach the information from left out data points is completely lost and it would be very hard to get a realistic picture of the uncertainties of the model. The more sophisticated sparse approximations, in contrast to just throwing out information, try to use the information present in the training data as well as possible without explicitly handling the full covariance matrix. They also aim to give a more realistic picture of the uncertainties present in the approximation. The more sophisticated sparse approximations include for example the *subset of regressors* approximation presented by Silverman (1985) and Wahba et al. (1999), the *deterministic training conditional* by Csató and Opper (2002) and Seeger et al. (2003) and the *Nyström approximation* proposed by Williams and Seeger (2001). The method used in this work is a *Fully independent training conditional* presented by Snelson and Ghahramani (2006) with a name *sparse pseudo-input Gaussian process*. A good overview and a unifying treatment of the different sparse approximations is given by Quinonero-Candela and Rasmussen (2005).

Before continuing to discussion of the sparse approximation used in this work, two results that are fundamental for the method are presented. First of them is the *matrix inversion lemma* or Woodbury, Sherman and Morrison formula:

Lemma 1 *Let* $\mathbf{Z}$ *be an* $n \times n$ *matrix,* $\mathbf{W}$ *an* $m \times m$ *matrix,* $\mathbf{U}$ *an* $n \times m$ *matrix and* $\mathbf{V}$ *an*

$n \times m$ matrix. If $\mathbf{Z}$ *and* $\mathbf{W}$ *are nonsingular, then* $\mathbf{UWV} + \mathbf{Z}$ *is nonsingular if and only if* $\mathbf{W}^{-1} + \mathbf{VZ}^{-1}\mathbf{U}$ *is nonsingular in which case*

$$(\mathbf{UWV} + \mathbf{Z})^{-1} = \mathbf{Z}^{-1} - \mathbf{Z}^{-1}\mathbf{U}(\mathbf{W}^{-1} + \mathbf{VZ}^{-1}\mathbf{U})^{-1}\mathbf{VZ}^{-1} \tag{4.23}$$

The advantage of the above lemma is obtained in the case of an $n \times n$ matrix $\mathbf{UWV} + \mathbf{Z}$ for which the inversion of $\mathbf{Z}$ is easy to construct. That is a case, for example, when $\mathbf{Z}$ is a diagonal matrix. In the sparse approximation used here, the inverse of $\mathbf{W}$ is known also and $\mathbf{Z}^{-1}\mathbf{U}$ is the transpose of $\mathbf{VZ}^{-1}$.

The other result is a *matrix determinant lemma* which states:

Lemma 2 *Let* $\mathbf{Z}$ *be an* $n \times n$ *matrix,* $\mathbf{W}$ *an* $m \times m$ *matrix,* $\mathbf{U}$ *an* $n \times m$ *matrix and* $\mathbf{V}$ *an* $n \times m$ *matrix. If* $\mathbf{Z}$ *and* $\mathbf{W}$ *are nonsingular, then*

$$|\mathbf{UWV} + \mathbf{Z}| = |\mathbf{Z}||\mathbf{W}||\mathbf{W}^{-1} + \mathbf{VZ}^{-1}\mathbf{U}|. \tag{4.24}$$

The advantage of the lemma results again if $\mathbf{Z}$ is of nice form. The proof for both of the lemmas is given, for example, by Harville (1997).

4.4.2 Fully independent training conditional

The fully independent training conditional (FITC) sparse approximation was first introduced by Snelson and Ghahramani (2006) with a name sparse pseudo-input Gaussian process. The name used here was given by Quinonero-Candela and Rasmussen (2005) in their unifying review of sparse Gaussian processes.

To start with the construction of FITC, the joint prior $p(\mathbf{f}, \mathbf{f}_*)$ in (4.6) is modified in a way, which will reduce the computational requirements from the predictive distribution (4.10). First the prior is rewritten using an additional set of *m inducing variables* $\mathbf{u} = [u_1, ..., u_m]^{\mathrm{T}}$ and forming the joint prior for all the latent and the inducing variables

$p(\mathbf{f}, \mathbf{f}_*, \mathbf{u})$. The inducing variables are latent variables of the Gaussian process, as well as $\mathbf{f}$ and $\mathbf{f}_*$, corresponding to a set of input locations $\mathbf{x}_u$ called *inducing inputs*. Due to the consistency property of Gaussian processes the original prior $p(\mathbf{f}, \mathbf{f}_*)$ can be recovered simply integrating out the inducing variables

$$p(\mathbf{f}, \mathbf{f}_*) = \int p(\mathbf{f}, \mathbf{f}_*, \mathbf{u}) d\,\mathbf{u} = \int p(\mathbf{f}, \mathbf{f}_* \mid \mathbf{u}) p(\mathbf{u}) d\,\mathbf{u}, \tag{4.25}$$

where $p(\mathbf{u}) = N\left(\mathbf{0}, \mathbf{K}_{\mathrm{u,u}}\right)$ is the inducing prior. The fundamental idea of most of the sparse approximations is to approximate the joint prior by assuming that $\mathbf{f}$ and $\mathbf{f}_*$ are conditionally independent given $\mathbf{u}$. This gives for the joint prior of latent values an approximation

$$p(\mathbf{f}, \mathbf{f}_*) \approx q(\mathbf{f}, \mathbf{f}_*) = \int q(\mathbf{f} \mid \mathbf{u}) q(\mathbf{f}_* \mid \mathbf{u}) p(\mathbf{u}) d\,\mathbf{u}, \tag{4.26}$$

where $q(\mathbf{f} \mid \mathbf{u})$ and $q(\mathbf{f}_* \mid \mathbf{u})$ are the *approximate inducing conditionals*. Here the latent variables $\mathbf{f}$ and $\mathbf{f}_*$ can communicate only through $\mathbf{u}$, which therefore *induces* the dependence's between the training and the test cases.

It is worth noting here that whereas the inducing variables $\mathbf{u}$ are always marginalized out in the predictive distribution, the choice of inducing inputs does leave an imprint on the final solution, as shown later in the figure 4.2. FITC approximation can thus be viewed as a standard Gaussian process with a particular non-stationary covariance function parametrized by the inducing inputs. The choice of the inducing inputs thus plays an important role in the goodness of the model.

Introducing a short hand notation $\mathbf{Q}_{a,b} = \mathbf{K}_{a,u}\mathbf{K}_{u,u}^{-1}\mathbf{K}_{u,b}$ the exact expressions for the training and test conditionals in (4.26) can be expressed as

$$p(\mathbf{f} \mid \mathbf{u}) = N(\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{u,u}^{-1}\,\mathbf{u}, \mathbf{K}_{\mathrm{f,f}} - \mathbf{Q}_{\mathrm{f,f}}) \tag{4.27}$$

$$p(\mathbf{f}_* \mid \mathbf{u}) = N(\mathbf{K}_{*,\mathrm{u}}\,\mathbf{K}_{u,u}^{-1}\,\mathbf{u}, \mathbf{K}_{*,*} - \mathbf{Q}_{*,*}). \tag{4.28}$$

Here $\mathbf{u}$ can be thought to play a role of special noise free observations in which case the expressions of the exact conditionals are special cases of the predictive distribution.

To complete the construction of FITC approximation the inducing conditionals are written

as

$$q_{\mathrm{FITC}}(\mathbf{f} \mid \mathbf{u}) = N(\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{u,u}^{-1}\,\mathbf{u},\, \mathrm{diag}\left[\mathbf{K}_{\mathrm{f,f}} - \mathbf{Q}_{\mathrm{f,f}}\right]) \tag{4.29}$$

$$q_{\mathrm{FITC}}(\mathbf{f}_* \mid \mathbf{u}) = N(\mathbf{K}_{*,\mathrm{u}}\,\mathbf{K}_{u,u}^{-1}\,\mathbf{u},\, \mathbf{K}_{*,*} - \mathbf{Q}_{*,*}) = f(\mathbf{f}_* \mid \mathbf{u}), \tag{4.30}$$

implying the effective joint prior of $\mathbf{f}$ and $\mathbf{f}_*$ as

$$p(\mathbf{f}, \mathbf{f}_*) = \mathcal{GP}\left(\mathbf{0}, \begin{bmatrix} \mathbf{Q}_{\mathrm{f,f}} - \mathrm{diag}\left[\mathbf{Q}_{\mathrm{f,f}} - \mathbf{K}_{\mathrm{f,f}}\right] & \mathbf{Q}_{\mathrm{f},*} \\ \mathbf{Q}_{*,\mathrm{f}} & \mathbf{K}_{*,*} \end{bmatrix}\right). \tag{4.31}$$

As discussed earlier in the context of a full Gaussian process the key equations in the GP inference are the posterior predictive distribution of a regression problem (4.10) and the energy functions in (4.13) and (4.16). Now in FITC approximation the covariance matrix $\mathbf{K}_{\mathrm{f,f}}$ in these equations is replaced by $\mathbf{Q}_{\mathrm{f,f}} - \mathrm{diag}\left[\mathbf{Q}_{\mathrm{f,f}} - \mathbf{K}_{\mathrm{f,f}}\right]$. By defining $\Lambda = \mathrm{diag}\left[\mathbf{K}_{\mathrm{f,f}} - \mathbf{Q}_{\mathrm{f,f}}\right]$ the predictive distribution and the energy function in regression problem can be written, respectively, as

$$p(\mathbf{f}_* \,|\mathbf{y}) = N\left(\mathbf{K}_{*,\mathrm{f}}(\mathbf{Q}_{\mathrm{f,f}} + \Lambda + \sigma^2_{\mathrm{noise}}\mathbf{I})^{-1}\mathbf{y},\, \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathrm{f}}(\mathbf{Q}_{\mathrm{f,f}} + \Lambda + \sigma^2_{\mathrm{noise}}\mathbf{I})^{-1}\,\mathbf{K}_{\mathrm{f},*}\right) \tag{4.32}$$

$$\begin{aligned} E = &-\frac{1}{2}\log\left|\mathbf{Q}_{\mathrm{f,f}} + \Lambda + \sigma^2_{\mathrm{noise}}\mathbf{I}\right| - \frac{1}{2}\mathbf{y}^{\mathrm{T}}\left(\mathbf{Q}_{\mathrm{f,f}} + \Lambda + \sigma^2_{\mathrm{noise}}\mathbf{I}\right)^{-1}\mathbf{y} \\ &- \frac{n}{2}\log(2\pi) - \log\left(p(\theta)\right). \end{aligned} \tag{4.33}$$

In the case of an arbitrary likelihood, it is not possible to solve analytically the predictive distribution. The noise term $\sigma^2_{\mathrm{noise}}\mathbf{I}$ is replaced and the likelihood is changed to the general $p(\mathbf{y}|g(\mathbf{f}))$ giving an energy function

$$\begin{aligned} E = &-\frac{1}{2}\log\left|\mathbf{Q}_{\mathrm{f,f}} + \Lambda\right| - \frac{1}{2}\mathbf{y}^{\mathrm{T}}\left(\mathbf{Q}_{\mathrm{f,f}} + \Lambda\right)^{-1}\mathbf{y} - \frac{n}{2}\log(2\pi) \\ &- \log\left(p(\mathbf{y} \mid g(\mathbf{f}))\right) - \log\left(p(\theta)\right). \end{aligned} \tag{4.34}$$

In a full Gaussian process, the computationally most prohibitive part of evaluations is the inversion of covariance matrix. Here the inversion of $\mathbf{Q}_{\mathrm{f,f}} + \Lambda$ or $\mathbf{Q}_{\mathrm{f,f}} + \Lambda + \sigma^2_{\mathrm{noise}}\mathbf{I}$ can be

transformed in a more efficient form using the matrix inversion lemma (4.23)

$$\left(\mathbf{Q}_{\mathrm{f,f}} + \Lambda\right)^{-1} = \Lambda^{-1} + \Lambda^{-1}\,\mathbf{K}_{\mathrm{f,u}}\left(\mathbf{K}_{\mathrm{u,u}} + \mathbf{K}_{\mathrm{u,f}}\,\Lambda^{-1}\,\mathbf{K}_{\mathrm{f,u}}\right)^{-1}\mathbf{K}_{\mathrm{u,f}}\,\Lambda^{-1}, \tag{4.35}$$

where the inversion of diagonal $n \times n$ matrix Λ can be transformed into an elementwise inversion of a vector diag $[\Lambda]$ and the only matrix inversion required is that of an $m \times m$ matrix $\left(\mathbf{K}_{\mathrm{u,u}} + \mathbf{K}_{\mathrm{u,f}}\,\Lambda^{-1}\,\mathbf{K}_{\mathrm{f,u}}\right)$. The computational cost is dominated by the matrix multiplication $\mathbf{K}_{\mathrm{u,f}}\,\Lambda^{-1}\,\mathbf{K}_{\mathrm{f,u}}$, which is $O(m^2 n)$ (Snelson and Ghahramani, 2006). Then, compared to the inversion of an $n \times n$ matrix of full GP the computational cost in FITC is reduced from $O(n^3)$ to $O(m^2 n)$. The determinant can also be evaluated more efficiently using the matrix determinant lemma (4.24) resulting in

$$\left|\mathbf{Q}_{\mathrm{f,f}} + \Lambda\right| = |\,\Lambda\,||\mathbf{K}_{u,u}^{-1}||\,\mathbf{K}_{\mathrm{u,u}} + \mathbf{K}_{\mathrm{u,f}}\,\Lambda^{-1}\,\mathbf{K}_{\mathrm{f,u}}\,|. \tag{4.36}$$

4.4.3 On the choice of the inducing inputs

Until now the choice of the inducing inputs $\mathbf{X}_u$ has not been considered. However, the choice of them is a crucial part of the model construction. Although the inducing variables $\mathbf{u}$ are marginalized out from the inducing conditionals, the choice of the inducing inputs does leave an imprint in the final inference, and thus the choice of them should be done with care. Traditionally the inducing inputs in sparse approximations are carefully chosen subset from the training or test inputs, but nothing in the construction of FITC approximation limits the choice on them.

Consider the predictive distribution (4.32) of the Gaussian process regression with FITC approximation. The predictive distribution is obtained from an analytic solution

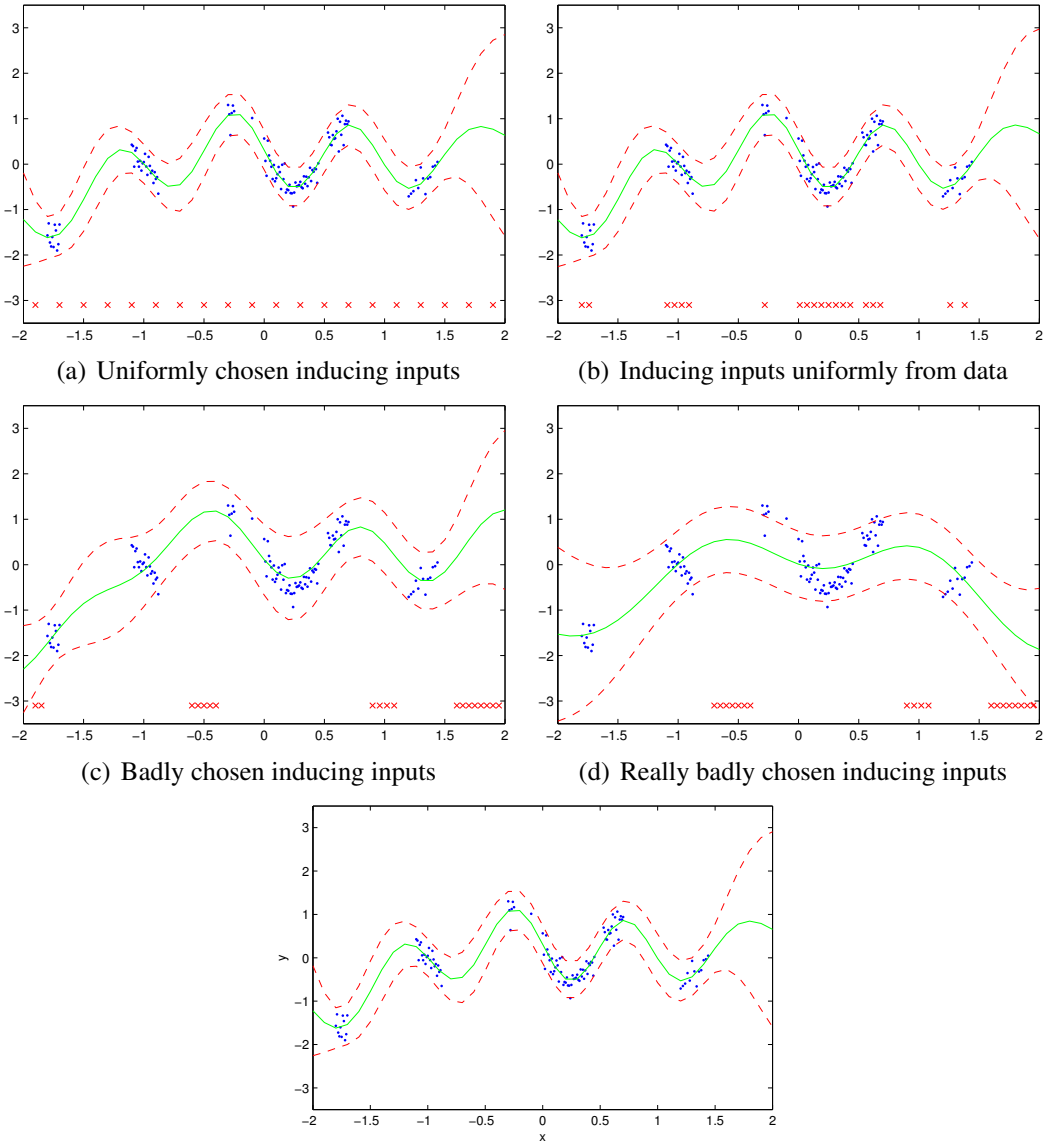
$$p(\mathbf{f}_* \,|\mathbf{y}) = N\left(\mathbf{K}_{*,\mathrm{f}}(\mathbf{Q}_{\mathrm{f,f}} + \Lambda + \sigma^2_{\mathrm{noise}}\mathbf{I})^{-1}\mathbf{y},\, \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathrm{f}}(\mathbf{Q}_{\mathrm{f,f}} + \Lambda + \sigma^2_{\mathrm{noise}}\mathbf{I})^{-1}\,\mathbf{K}_{\mathrm{f},*}\right),$$

where $\mathbf{Q}_{\mathrm{f,f}} = \mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{u,u}^{-1}\,\mathbf{K}_{\mathrm{u,f}}$ and the elements of covariance matrices are given as a function of inputs $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. The characteristic of covariance functions, as discussed in section 4.3, is that the further apart two points are from each other the smaller the

covariance between them is and thus the less they have influence on each others. The distance, after which the covariance between two points is negligible, is governed by the length scale. Since $\mathbf{Q}_{\mathrm{f,f}}$ and Λ are functions of $\mathbf{K}_{\mathrm{f,u}}$ and $\mathbf{K}_{\mathrm{u,f}}$, it can be seen from above that the posterior expectation of FITC approximation $\mathbb{E}[\mathbf{f}_*] = \mathbf{K}_{*,\mathrm{f}}(\mathbf{Q}_{\mathrm{f,f}} + \Lambda + \sigma^2_{\mathrm{noise}}\mathbf{I})^{-1}\mathbf{y}$ approaches the prior expectation $\mathbb{E}[\mathbf{f}_*] = \mathbf{K}_{*,\mathrm{f}}(\mathrm{diag}\left[\mathbf{K}_{\mathrm{f,f}}\right] + \sigma^2_{\mathrm{noise}}\mathbf{I})^{-1}\mathbf{y}$ as the distance between the inducing and the training inputs increases, To be precise the potteries approaches only an approximate prior, because there is only diagonal of $\mathbf{K}_{\mathrm{f,f}}$. Similarly the posterior covariance approaches the prior covariance. Also the opposite holds, as the number and location of inducing inputs approaches those of training set inputs the solution approaches the full model.

As well as the increase in the distance between the inducing and the data inputs moves the predictions towards the prior, the prior dominates also the posterior inference of the parameter values in the same case. Thus, there may first be a problem in finding the posterior of parameters and then in finding the posterior predictions with those parameters. The influence of the inducing inputs on the posterior inference is demonstrated in the figure 4.2, where a FITC sparse Gaussian process is applied for the same regression problem data as in the figure 4.1. In each case there is a same number of inducing inputs. The posterior mean of length scale in full GP is approximately $l = 0.6$. The upper two cases represent solutions, where the inducing inputs were chosen uniformly from the area spanned by the data and uniformly from the data inputs. The expectation of $\mathbf{f}_*$ is rather similar in both cases. The dashed red lines represent the 2σ, which is two times the standard deviation predicted by the model. In the lower cases it is seen that the inducing inputs are too far away from the data points for the model to fit in the data. In this case the data is explained in growing amount with the variance, which is seen in the wider distance between $\mathbf{f} \pm 2\sigma$ lines.

Snelson and Ghahramani (2006) choose the inducing inputs by maximizing the marginal likelihood with respect to a fixed number of them. In this work the inducing inputs are chosen uniformly from the data inputs. A full Bayesian approach would be to marginalize out the inducing inputs with, for example, Monte Carlo integration.



(a) Uniformly chosen inducing inputs

(b) Inducing inputs uniformly from data

(c) Badly chosen inducing inputs

(d) Really badly chosen inducing inputs

(e) A full GP for comparison

Figure 4.2: **An example of GP regression with FITC sparse approximation**. The data points are marked with blue dots, the *green* line represents the output f_* of trained GP, dashed red lines are the $f_* \pm 2\sigma$ and red crosses represent the locations of the inducing inputs. The σ is the standard deviation predicted by the model. The number of the inducing inputs is same in all the figures.

Chapter 5

Constructing the model

The focus of this work is to construct a full Bayesian model for finding possible spatial variations in the death rates of chronic diseases. The data available is a point referenced health data with various covariates. The approach to the study problem follows a generic hierarchical three level model with the Poisson likelihood and a sparse log Gaussian process prior. Gaussian process should be a reasonable choice to construct the intensity surface for the relative risk, since the surface is naturally smoothed by the process and the spatial correlations between areas can be included in an explicit and natural way into the model via a correlation function. The hyperprior is defined by half-Students'-t distribution to allow a priori small and moderate size process variation for the intensity surface.

The chapter starts by describing the case data under study, after which the model construction is treated in more detail. At the end of the chapter the model is placed under model criticism to analyze the restrictions and faults in it.

5.1 Data sets studied

For testing the FITC sparse approximation there were four different study sets constructed from the data available. The sets of data consisted of the mortality due to two different

diseases, *cerebral vascular diseases* and *alcohol-related diseases*, in the time interval 1995-1999. The data sets were studied with lattice resolutions of 20km × 20km and 10km × 10km resulting in 915 and 3193 data points respectively. The cerebral vascular diseases comprised roughly 18 000 deaths and the alcohol-related diseases about 5200 deaths.

5.2 Sparse log Gaussian process model

The model constructed in this work follows the general approach discussed in section 3.2.4. The data is aggregated into areas A_i with co-ordinates $(x_{i,1}, x_{i,2})$ and consist of information about the number of the death cases and the background population, and the explanatory covariates for both mortality and background data. The likelihood is Poisson with mean $E_i \mu_i$, where the standardized expected number of deaths E_i is evaluated using an age, gender and scholarly degree standardization as discussed in section 3.2.5. The log relative risk is given a Gaussian process prior with zero mean. The complete model until second level prior is

$$\mathbf{Y} \sim \text{Poisson}(\mathbf{E}\mu) \tag{5.1}$$

$$\log(\mu) = \mathbf{f}(\mathbf{x}_i, \mathbf{x}_j) \sim \mathcal{GP}(\mathbf{0}, \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)). \tag{5.2}$$

The drawback of GP is the computational burden of the required covariance matrix inversion. The computation time becomes prohibitive as the data amount increases up to around a few thousand of cases, limiting the study either to very small areas or a sparsely populated grid. To overcome the computational limitations the GP is given an FITC sparse approximation, from which the name *sparse log Gaussian process*.

The model here is similar to the log Gaussian Cox processes discussed, for example, by Møller et al. (1998); Beneš et al. (2002). The main difference between the Cox processes and the approach in this work is that the Cox process is defined strictly only for spatial point processes, in which the co-ordinates of data are random variables constructed by the Cox process. Here the co-ordinates of point referenced data are fixed, not random. The

data can be aggregated into areas of various sizes and the same model can be used for the resulting areal data using the co-ordinates of sub-regions as point co-ordinates.

5.3 Prior for covariance function parameters

The covariance functions used in the work are a squared exponential, an exponential, a Mátern $\nu = 3/2$ and a Mátern $\nu = 5/2$ discussed in section 4.3. It is a priori plausible that the process variance is zero or very small and thus the prior for covariance function parameters should be such that it enables both the length-scale l and the magnitude σ^2 to reach zero. The prior should also allow, especially for the length scale, higher values reflecting to correlating points far apart. To obtain these characteristics the covariance function parameters are both given a half-Students't prior (Gelman, 2006)

$$p(l|\nu = 1, A = 4) \propto \begin{cases} 0 & \text{if } l < 0, \\ \left(1 + \frac{1}{\nu}\left(\frac{l}{A}\right)^2\right)^{-(\nu+1)/2} & \text{otherwise} \end{cases} \tag{5.3}$$

$$p(\sigma_.^2|\nu = 0.3, A = 4) \propto \begin{cases} 0 & \text{if } l < 0, \\ \left(1 + \frac{1}{\nu}\left(\frac{\sigma_.^2}{A}\right)^2\right)^{-(\nu+1)/2} & \text{otherwise,} \end{cases} \tag{5.4}$$

where A is the scale and ν the degrees of freedom. The prior distributions are shown in the figure 5.1

5.4 Inducing inputs

The inducing inputs are chosen uniformly from the data. In the case of 20km×20km lattice with 915 data points every other data input is chosen resulting in 221 inducing inputs and in the 10km×10km lattice 238 inducing inputs are chosen by taking every fourth of 3193 data inputs. The distance between inducing inputs is in both cases 40km and the maximum distance from a data input to the nearest inducing input is 20 kilometers. The choice of the inducing inputs is shown in the figure 5.2.

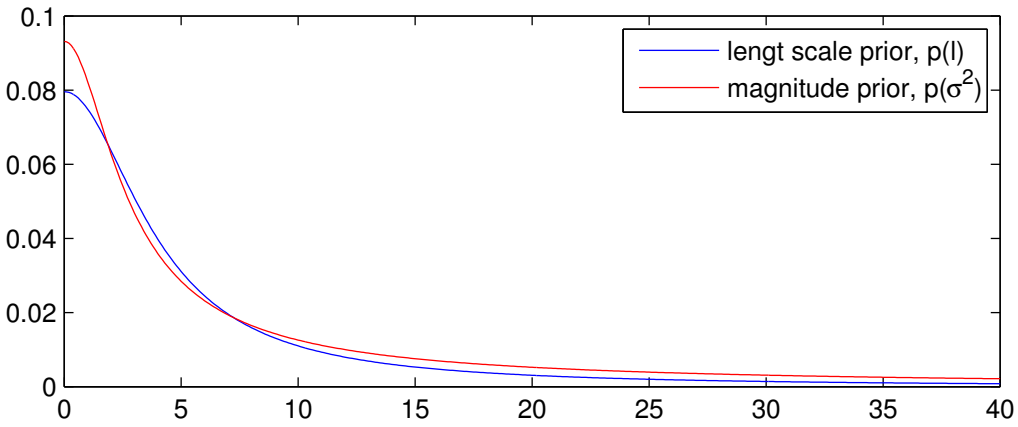


Figure 5.1: **The prior distribution for length scale and magnitude of covariance function**.

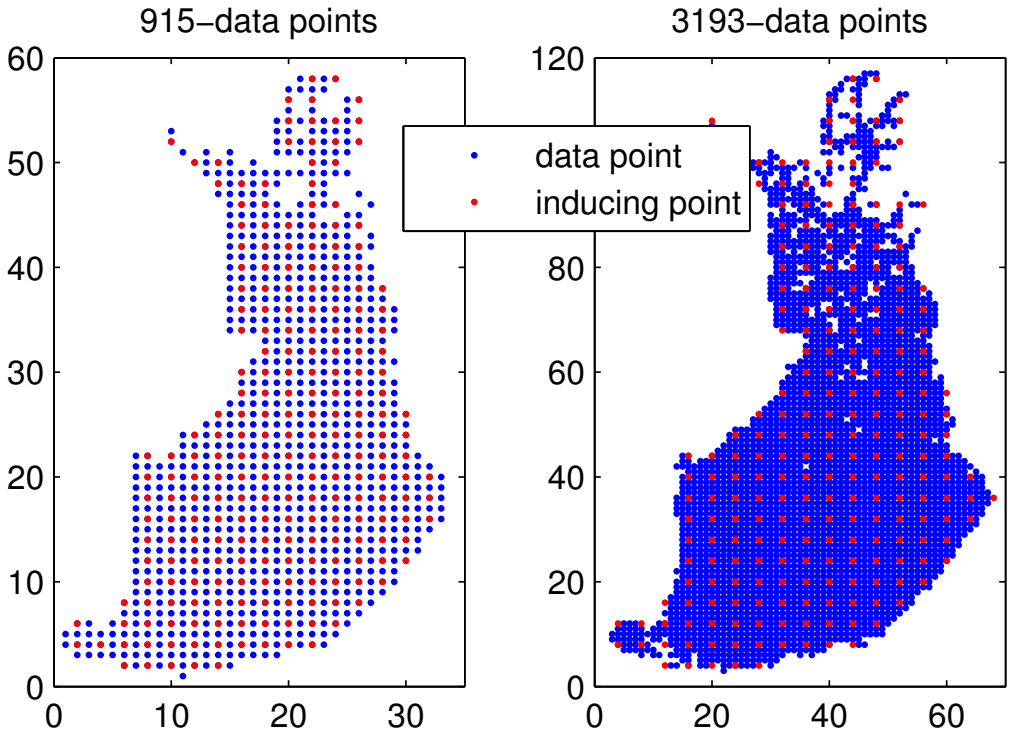


Figure 5.2: **The inducing inputs in the case study problems**. The inducing inputs were chosen by taking uniformly every other data point in the case of 915 data points, *on the left*, and every fourth data point in the case of 3193 data points, *on the right*. From the pictures it is also seen how sparsely populated country Finland is on the north. The white areas contain cells with no population.

5.5 Model criticism

Poisson distribution is a widely used approximation for the likelihood of the death rate. Due to the characteristic that the mean and the variance of Poisson distribution are the same parameter, Poisson distribution can be considered as a good approximation of the underlying binomial distribution in the cases of a large population size and a small number of diseases cases. This is the general situation for example in the densely populated countries and cities of middle Europe. However, in Finland and especially in the northern and eastern parts of Finland, the population sizes are rather small, which results in high uncertainty in the expected death rate and thus the noise allowed by the Poisson distribution may be too small, resulting in an extra-Poisson variation.

In the areas, where the Poisson likelihood is too strict, the posterior of the relative risk may obtain unreasonable large values and thus result in a false interpretation of a spatial effect. The background population is obtained from the census surveys conducted every five years and the changes in population during that time may be rather big. Especially the migration from the sparsely populated rural areas into the big cities increases the already high uncertainty of the expected death rate in those areas. The problem of extra-Poisson variation is discussed in brief for example by Diggle (2001).

Gaussian process should be a reasonable prior for $\log(\mu)$, since the surface of the log relative risk is naturally smoothed by the process and the spatial correlations between areas can be included in an explicit and natural way into the model via a correlation function. Gaussian distribution is symmetric around its mean, which is not necessarily the case with the distribution of μ. However the log transformation of μ reduces the possible non-symmetry and $\log(\mu)$ is also more likely to be Gaussian distributed with mean zero than μ with mean one.

The inducing inputs of FITC approximation were chosen uniformly over data inputs. As discussed in section 4.4.3 this results in rather invariable variance estimate in all areas. By choosing more inducing inputs from the areas with large population density and less in the areas of small population density the uncertainty of the predictions could be reduced

in the areas where also the data uncertainty is smaller. The number of the inducing inputs was chosen so that the distance from a data input to the nearest inducing input is not too big. This distance is 20km at maximum, which means that spatial effects with length scale lot smaller than 20km can not be found. However, in the case of 20km$\times$20km lattice that kind of spatial effects can not be found with full model either, since the distance between two data inputs is also 20km at minimum. In the denser 10km$\times$10km lattice the choice of inducing inputs may already have effect on how fast varying phenomenon the model can adapt. The best approach in choosing the inducing inputs would be to sample the number and locations of them with for example Reversible Jump Markov Chain Monte Carlo method (Green, 1995).

Chapter 6

Computational methods

Computational methods play an essential role in applications of all statistical methods. In Bayesian analysis the integrals resulting from the marginalization principle can not in general be solved analytically and thus they are given either analytic or numerical approximations. The approximations require both novel methods and computational power. In this work, not only the integrals of Bayesian analysis are computationally demanding, but also Gaussian processes lead to time consuming calculations. In particular the inverse of the covariance function and the matrix multiplications need to be conducted with care.

The advantage of FITC approximation is that there is no need to invert or construct any $n \times n$ matrix in the computations. This chapter begins by introducing the implementation environment and discussing some tricks used to avoid $n \times n$ matrices in calculations. Next, the discussion is given about the Markov chain Monte Carlo methods, which are used to conduct the integration over the nuisance parameters of the model. After introducing Markov chains and the iterative methods to construct them, the discussion is continued with transformations of parameters and in particular of the transformation of the latent values with respect to their approximate posterior variance. The chapter ends with derivation of the gradients of an energy function.

6.1 Implementation issues

6.1.1 Implementation environment

The model and the methods discussed above are implemented in Matlab 7.* environment as a part of a Gaussian processes toolbox. Main parts of the toolbox were written during the work on the thesis and at the moment the first version of the toolbox is usable. The toolbox follows the idea and uses some of the code of MCMCstuff toolbox (available in the Internet at http://www.lce.hut.fi/research/mm/mcmcstuff/), which is a collection of Matlab functions for Bayesian inference with MCMC methods. The two toolboxes are not compatible with each others and the future objective is to publish also the new toolbox in the Internet, and thus provide also a reference implementation for the methods discussed here.

Matlab provides an efficient environment to implement and test new methods due to its easy to use syntax and wide variety of ready made toolboxes. However, Matlab functions are designed to handle matrices with general structure, and thus they are not the most efficient choice for manipulating large matrices with known properties. In Gaussian processes, for example, a covariance matrix is both symmetric and positive definite, which enables the use of more efficient algorithms for computing matrix products, determinants and inversions. For a treatment of algorithms for matrix evaluations see for example the treatment of Golub and van Loan (1996).

At the moment, this work is implemented in Matlab and the matrix evaluations in the code are optimized for fast performance with the tools provided by Matlab. Special care is taken in the matrix computations in FITC approximation, which are conducted in a manner that all matrices of size $n \times n$ are avoided. The Matlab version 7.* was chosen for implementation environment because of the nested function property provided by it. The nested functions provide an easy way to share common information between related functions.

6.1.2 About computations with matrices and vectors

As discussed above in the context of Gaussian processes the most time consuming part of the implementation is the inversion of the full covariance matrix, which needs a time proportional to $O(n^3)$. However, other matrix evaluations such as multiplications and determinants of matrices may become computationally prohibitive as well. Unnecessarily large matrices should be avoided also for saving memory, since the memory for storing a matrix is proportional to $O(n^2)$.

In general, the computational cost and memory requirements for solving matrix problems can be reduced significantly, if the implementation is conducted with care. The order of computations, saving intermediate results, reuse of storage variables and taking advantage of the possible known structure of matrix are examples of basic tricks that should be used in the programming. In the case of symmetric covariance matrix a practical operator is also the Cholesky decomposition:

Definition 2 *The Cholesky decomposition of a symmetric, positive definite matrix* $\mathbf{A}$ *decomposes A into a product of a lower triangular matrix* $\mathbf{L} = chol\,[\mathbf{A}]$ *and its transpose*

$$\mathbf{L}\mathbf{L}^T = \mathbf{A}. \tag{6.1}$$

Cholesky decomposition is numerically very stable and it is useful in many calculations involving symmetric matrices. An example of using Cholesky decomposition is the evaluation of $\mathbf{y}^{\mathrm{T}}\,{\mathbf{K}_{\mathrm{f,f}}}^{-1}\,\mathbf{y}$, needed for example in the evaluation of an energy function (4.16). This can be evaluated efficiently solving first $\mathbf{b} = \mathrm{chol}\left[\mathbf{K}_{\mathrm{f,f}}\right] \setminus \mathbf{y}$ and then evaluation $\mathbf{b}^{\mathrm{T}}\mathbf{b}$, where the notation $\mathbf{A} \setminus \mathbf{x}$ is the vector $\mathbf{x}$ that solves the $\mathbf{A}\mathbf{x} = \mathbf{b}$, also called forward substitution. The computation of Cholesky decomposition takes time $n^3/6$ and the forward substitution of time $n^2/2$. The Cholesky decomposition can also be used to evaluate the determinant of a symmetric and positive definite matrix as follows

$$|\mathbf{K}| = \prod_{i=1}^{n} \mathbf{L}_{ii}^2, \tag{6.2}$$

where $\mathbf{L}$ is the Cholesky decomposition of $\mathbf{K}$.

The last example of the use of Cholesky is the construction of the diagonal Λ matrix in equation (4.35) without forming the full $n \times n$ matrix $\mathbf{Q}_{\mathrm{f,f}}$. The diagonal elements of $\mathbf{K}_{\mathrm{f,f}}$ can be easily obtained from the covariance function $k(\mathbf{x}_i, \mathbf{x}_i)$ and diag $\left[\mathbf{Q}_{\mathrm{f,f}}\right]$ can be constructed by first evaluating

$$\mathbf{B} = (\mathrm{chol}(\mathbf{K}_{\mathrm{u,u}})^{-1})^{\mathrm{T}}\, \mathbf{K}_{\mathrm{u,f}} \tag{6.3}$$

where the transpose is needed because the inversion is taken after Cholesky decomposition. From this the diagonals of $\mathbf{Q}_{\mathrm{f,f}}$ are obtained as $\mathbf{Q}_{\mathrm{f,f}}(j, j) = \Sigma_i b_{ij}^2$, where b_{ij} is the ijth element of $\mathbf{B}$, and the diagonal elements of Λ are obtained from $\Lambda(i, i) = k(\mathbf{x}_i, \mathbf{x}_i) - \mathbf{Q}_{\mathrm{f,f}}(i, i)$. The diagonal elements can then be stored in a vector of length n, which requires a memory proportional to only $O(n)$. The matrix operations can also be conducted faster with the vector of diagonal elements than with the full diagonal matrix. For example the multiplication $\Lambda\, \mathbf{K}_{\mathrm{f,u}}$ corresponds to multiplying the rows $\mathbf{K}_{\mathrm{f,u}}(i, \cdot)$, $i = 1, ..., n$, with the respective diagonal elements Λ_{ii}.

6.2 Markov chain Monte Carlo methods

Bayesian analysis usually results in complex integrals that are not analytically tractable. As discussed in section 3.1.3 there are wide variety of approximative methods, of which the numerical *Markov chain Monte Carlo* (MCMC) methods are considered here (e.g. Gilks et al., 1996). In Monte Carlo integration for example the expectations of the form

$$\mathbb{E}\left[f\right] = \int f(\mathbf{x}, \theta)\, p(\theta) d\theta, \tag{6.4}$$

are approximated, using a sample of values $\theta^{(t)}$ drawn from the distribution $p(\theta)$, by

$$\mathbb{E}\left[f\right] \approx \frac{1}{N} \sum_{t=1}^{N} f(\mathbf{x}\,|\theta^{(t)}, M). \tag{6.5}$$

Thus the population mean of $f(\mathbf{x}, \theta)$ is estimated by a sample mean. When the samples $\theta^{(t)}$ are independent, the laws of large numbers ensure that the approximation can be made as accurate as desired by increasing the sample size N (e.g Gilks et al., 1996).

The samples $\theta^{(t)}$ can be drawn from the desired distribution with MCMC methods. A Markov chain is defined as a sequence of random variables satisfying the Markov property:

Definition 3 *A stochastic process has the Markov property if the conditional probability distribution of future states of the process, given the present and past states, depends only on the current state of the process, that is*

$$\Pr\left[X^{(t+1)} = x^{(t+1)} \mid X^{(0)} = x^{(0)}, ..., X^{(t)} = x^{(t)}\right] = \Pr\left[X^{(t+1)} = x^{(t+1)} \mid X^{(t)} = x^{(t)}\right]. \tag{6.6}$$

The probability (6.6) is defined as the *transition probability* of a Markov chain.

In order to produce a chain of samples that converge to a *stationary distribution*, three important properties have to be satisfied. First the chain has to be *irreducible*. That is, every state of the process has to be acceptable from every other state with positive probability and in some number of iterations. Secondly the chain needs to be *aperiodic*. This prevents the Markov chain from oscillating between different sets of states in a regular periodic movement. And most importantly, the chain has to be *positive recurrent*. In positive recurrent chain the expected return time to any state is finite. A positive recurrent and aperiodic chain is also called *ergodic*. The fundamental result in constructing Markov chains is the ergodic theorem, which assures for an ergodic chain a sure converge towards a unique stationary distribution in the limit of infinite long chain (Roberts, 1996).

Markov chain Monte Carlo methods are a class of algorithms for sampling from a probability distribution based on constructing a Markov chain with the desired distribution as its stationary distribution. The basic building block in constructing such an algorithm is to satisfy the ergodicity of the chain. However, even the ergodic theorem ensures the convergence towards the stationary distribution, it does not offer any information about the time

needed for convergence. Moreover, most of the time it is infeasible to draw independent samples from the Markov chain, resulting in autocorrelation between nearby samples. As a result it is essential to have tools to monitor the convergence and the independence of the samples as discussed in the section 6.2.4.

There are a wide variety of MCMC sampling algorithms presented in the literature. The efficiency of the algorithm depends highly on the problem at hand and, thus, a method that works well in one problem can fail in the other. In this section, the three sampling algorithms used in the work are introduced and discussed shortly. The theoretical background and a more extensive discussion of MCMC methods are given, for example, by (Gilks et al., 1996; Neal, 1996; Gelman et al., 2004; Nabney, 2001).

6.2.1 Metropolis Hastings algorithm

The Metropolis-Hastings algorithm is a generalization of a random walk Metropolis algorithm. The Metropolis algorithm utilizes a proposal distribution, from which a new candidate state is generated and, then either accepted or rejected based on the probability density ratio between the proposed and the current states. The Metropolis algorithm is an adaptation of a random walk, where the ergodicity of the chain is ensured by the acceptance rule between the old and the new states.

The proposal distribution of the Metropolis algorithm has to be symmetric to both directions, the moves from the current state to a new and from the new state to the current. In many cases the symmetry requirement leads to an inefficient sampling and Metropolis-Hastings algorithm is planned to overcome this limitation. In Metropolis-Hastings the acceptance rule is based on both the ratio of proposal distributions and the ratio of probability densities. The algorithm is as follows.

1. Draw a starting point $\theta^{(0)}$, for which $p(\theta^{(0)}|D) > 0$
2. for $t = 1, 2, ...$
   (a) Sample a proposal $\theta^{(*)}$ from proposal distribution $J_t(\theta^{(*)}|\theta^{(t-1)})$

(b) Calculate the acceptance ratio

$$\alpha = \min\left(1, \frac{p(\theta^{(*)}|D)J_t(\theta^{(t-1)}|\theta^{(*)})}{p(\theta^{(t-1)}|D)J_t(\theta^{(*)}|\theta^{(t-1)})}\right) \tag{6.7}$$

3. Set

$$\theta^t = \begin{cases} \theta^{(*)} & \text{with probability } \alpha \\ \theta^{(t-1)} & \text{otherwise.} \end{cases}$$

The Metropolis-Hastings algorithm is easy to construct and rather efficient for distributions with a low dimensionality. The essential part is the proposal distribution which highly influences the effectiveness of the algorithm. The optimal rejection rate is 0.56 for one dimension and 0.77 when many parameters are updated at once. Metropolis-Hastings faces its limitations in a case of a high dimensional and/or a heavily skewed distribution. For more detailed treatment and proof of ergodicity see the discussion of Gelman et al. (2004).

6.2.2 Gibbs sampling

Gibbs sampling (e.g Gelman et al., 2004) is one of the basic algorithms among the Markov chain Monte Carlo methods. The method is particularly useful for sampling multi-dimensional distributions, if the sampling of joint distribution of the variables is not directly feasible but there is an effective sampling algorithm to sample from the conditional distributions of each variable or a subset of variables. The basic idea of the sampler is to generate samples from the joint distribution by cycling through all the parameters and draw one parameter or one subset of parameters in turn, conditioned on the current values of all the others. Now, suppose a parameter vector $\theta = \{\theta_1, \theta_2, ..., \theta_n\}$ with n components or subvectors. At iteration t the Gibbs sampler cycles through all the n components of θ and draws a new value $\theta_j^{(t)}$ for each parameter from its conditional distribution given all the other parameters at their current values

$$p\left(\theta_j^{(t)}|\theta_1^{(t)}, \theta_2^{(t)}, ..., \theta_{j-1}^{(t)}, \theta_{j+1}^{(t-1)}, ..., \theta_n^{(t-1)}\right). \tag{6.8}$$

Therefore the conditional sampling at each iteration is performed n times and new sampled values are immediately used.

6.2.3 Hybrid Monte Carlo

The efficiency of Metropolis and Metropolis-Hastings algorithm depends highly on the goodness of the proposal distribution. If the parameter space sampled is high dimensional and the different components of parameters have a different size of a variance, or worse, are highly correlated it is hard to construct an efficient proposal distribution. If the variance of the proposal distribution is too large the components with a low variance lead to high rejection rate and in the case of a too low variance the autocorrelation time increases. Also the time needed for the algorithm to move from one end of the distribution to the other is rather long. The phenomenon is discussed more in section 6.3.2. To overcome these difficulties, hybrid Monte Carlo (HMC) algorithm was proposed by Duane et al. (1987). The discussion of the method follows the treatment of Neal (1996).

Hybrid Monte Carlo algorithm uses the basic idea of Metropolis-Hastings algorithm so that the candidate states are generated by dynamical simulations in a phase space of *position* and *momentum* variables. The position variable q, with n real valued components q_i, corresponds to the sampled parameters θ, and for every component of position there is a momentum k_i related to it.

The probability density of position may be written in a canonical form as

$$P(q) \propto \exp(-E(q)), \tag{6.9}$$

where $E(q)$ is the potential energy function of q. Hence, for a non-zero probability density function $P(q)$, it is possible to define an energy $E(q) = -\log(P(q))$. The energy function may be, for example, the log-posterior cost function 3.7.

In addition to energy function a kinetic energy $K(k)$ due to the momentum is needed to utilize the dynamical methods. Having both energy functions a Hamiltonian function that

gives the total energy of the system may be constructed as $\mathcal{H}(q, k) = E(q) + K(k)$, after which the canonical distribution over the phase space is obtained from

$$P(q, p) \propto \exp(-\mathcal{H}(q, k)). \tag{6.10}$$

The sampling in HMC is split into two sub-tasks 1) sampling for the values of q and k by the dynamical simulation with a fixed total energy, $\mathcal{H}(q, p)$, (dynamic sampling) and 2) sampling energy states $\mathcal{H}$ using the Gibbs method (stochastic sampling). Then, by altering the deterministic dynamical simulation and the stochastic energy level sampling an ergodic Markov chain from the desired distribution can be produced.

The dynamic sampling is conducted by moving from the starting position (q, p) to the new position $(q*, p*)$ according to the Hamiltonian dynamics of the system. The Hamiltonian dynamics for fixed $\mathcal{H}(q, p)$ could be followed exactly by integrating along the path $(q(t), k(t))$, where t denotes time. This, however, is not possible in practice and thus the dynamics must be simulated by discretized time steps. In a leapfrog democratization starting from q and k at time t an approximation to the position $q*$ and momentum k, at time $t + \delta t$ is obtained as follows

$$k_i(t + \frac{\delta t}{2}) = k_i(t) - \frac{\delta t}{2}\frac{\partial E}{\partial q_i}(q(t)) \tag{6.11}$$

$$q_i(t + \delta t) = q_i(t) + \delta t \frac{k_i(t + \frac{\delta t}{2})}{m_i} \tag{6.12}$$

$$k_i(t + \delta t) = k_i(t + \frac{\delta t}{2}) - \frac{\delta t}{2}\frac{\partial E}{\partial q_i}(q(+\delta t)), \tag{6.13}$$

where m_i represents the mass associated with the component i. In an exact dynamics the total energy of the Hamiltonian system would remain constant, but the simulation with discretized timesteps causes error in the total energy, that is $\mathcal{H}(q*, p*) \neq \mathcal{H}(q, p)$. The bias resulting from the error is, however, eliminated by the occasional rejections based on the canonical distribution (6.10). The dynamical simulation may be summarized as

1. Starting from the current state, (q, k), perform L leapfrog steps with a step size δt to reach the state $(q*, p*)$.

2. Negate the momentum variables producing the state $(q', k') = (q*, -k*)$

3. Regard (q', k') as a candidate for the next state, accepting it with probability $\min(1, \exp(-(H(q', k') - H(q, k)))$.

A crucial part of the HMC sampling are the choice of the time step size δt, also called a *step size adjustment factor*, and the number of steps taken. The parameters should be tuned so that 5-15% of all the candidates are rejected, the portion of rejected samples is called a *rejection rate*.

A key part of HMC are the gradients with respect to the position variables. The gradient evaluation is discussed more in the section 6.4. A special case of HMC sampling is the Langevin-Hastings algorithm, (e.g. Møller and Waagepetersen, 2003) in which only one leapfrog step is used to move from current position to a new one. A more complete treatment concerning the hybrid Monte Carlo method is given by Neal (1996) and Neal (1993).

Hybrid Monte Carlo with persistence

In the hybrid Monte Carlo with *persistence* for the momentum the momentum is replaced only partially between trajectories. This causes that the motion will tend to persist in largely the same direction from step to step. The partial replacement of the momentum variables is made as following

$$k_i^{\text{new}} = \lambda_{\text{pers}} k_i + (1 - \lambda_{\text{pers}}^2)^{1/2} k_i^{\text{Gibs}}, \tag{6.14}$$

where the persistence parameter λ_{pers} adjust how much of the old momentum is replaced and k_i^{Gibs} is the momentum component obtained from Gibbs sampling. The persistence property may be useful, if the momentum would otherwise be changed too much considering the old position parameter. As well, if the parameters θ sampled with HMC are strongly dependent on other parameters that change between trajectories, the persistence property may help the sample chain of θ to explore the distribution space faster. For more

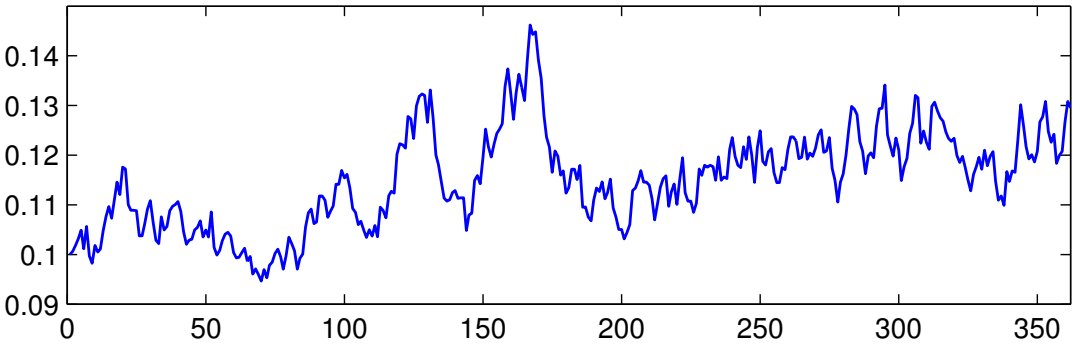
complete discussion about persistence see the treatment of Neal (1996).

6.2.4 Monitoring convergence

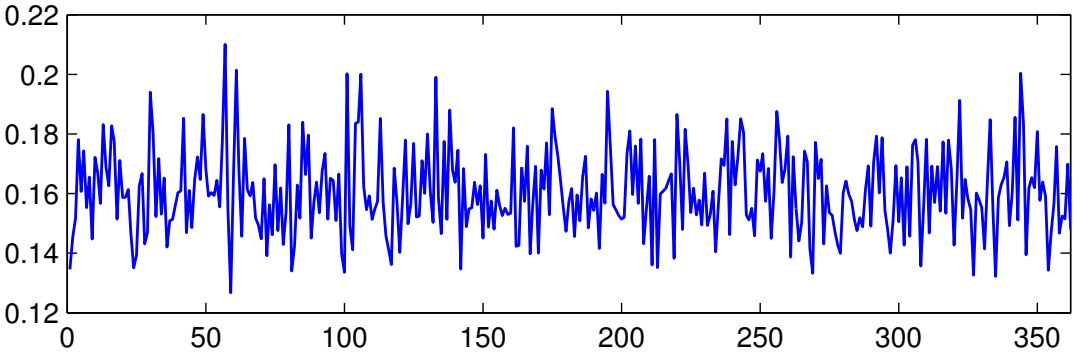
The use of MCMC samples for a posterior inference is based on the assumption that the samples are independently and identically distributed samples from a desired distribution. If the sampler satisfy the needed postulates, the theory of Markov chains ensures the convergence of chain to the right distribution in the limit of infinite long chain. However, it is possible to use only a finite number of samples as an approximation of the real distribution, in which case the convergence may still be incomplete. As well the samples drawn are correlated to the nearby samples or the sample chain may be stuck in a local mode. Resulting from the above uncertainties, it is compulsory to verify the goodness of the sample chain and to be able to point out possible convergence problems.

There are two important characteristic numbers to measure the goodness of sample chain, *burn-in* and *autocorrelation time*. Burn-in represents the time needed from the sample chain to reach the approximate equilibrium and autocorrelation time defines the distance between two nearest uncorrelated samples. For example, if autocorrelation time of chain is τ the first sample after $\theta^{(t)}$ not correlated to it is $\theta^{(t+\tau)}$. The first step in monitoring the convergence and the correlation is to inspect it visually. In the figure 6.1 there is an example of heavily correlating sample chain that has not converged and a converged sample chain with no correlation. In the literature there are discussed a number of computational methods to approximate the burn-in and the autocorrelation time.

As the very first step of verifying the chain, the number of burn-in samples should be removed. Approximations for the burn-in are obtained by using, for example, *potential scale reduction factor* (PSRF) (Brooks and Gelman, 1998). PSRF test estimates when two or more sample chains started from different points are from the same distributions by comparing the between variation and within variation. The PSRF test can be used also for one sample chain, in which case the factor is calculated between the first and last parts, for example, first and last third of the chain. However, use of only one sample chain



(a) A non-converged sample chain with autocorrelation $\tau = 284$.



(b) A converged sample chain with no autocorrelation.

Figure 6.1: **An example of a correlating non-converged and a converged non-correlating sample chain**.

produces over-optimistic results.

After removing the burn-in the autocorrelation time can be estimated in various methods. The one used in this work is *Geyer's initial monotone sequence estimator* (Geyer, 1992). After the autocorrelation time is determined the sample chain should be thinned by taking in only every τ'th sample from the original sample chain. An alternative to thinning is *batching* (e.g. Neal, 1993). The sample chain is divided evenly into batches of the same size and the mean or median of each batch is evaluated. These can then be handled as (quasi-) independent samples.

The convergence of the sample chain should be checked before and after thinning. After removing the burn-in the estimate for autocorrelation time is more reliable and as well the convergence estimate is more reliable after thinning. A *Kolmogorov-Smirnov test* (Robert and Casella, 2004) can be used as an additional test against non-convergence. The test

assumes independence of samples and thus is sensitive to the auto-correlation.

The sampling parameters and the length of the sample chain must be selected so that the test values from above tests are acceptable. The autocorrelation time should not be much more than 5% of all samples; otherwise the estimation of it is unreliable. The number of samples divided by the autocorrelation time tells roughly the effective sample size.

6.3 Transformation of latent values

6.3.1 Transformation of variables

It is common to transform a probability distribution from one parametrization to another (e.g. Gelman et al., 2004). For example the parameter space, in which the model is defined, may not always be the optimal for computational purposes, as MCMC sampling, or it may be easier to construct a prior structure for a transformed parameter or parameters, as μ in (5.2). In some applications it may also be useful to transform between parameter spaces of different dimensionality (Green, 1995). In this work a log parametrization of the covariance function parameters and a transformation of the latent values of Gaussian process with their approximate posterior variance play an important role in the implementation. Here, the aim of the transformations is to reduce the dependency of parameters on each others to make their sampling easier. Some basic results for a probability density on a transformed space are discussed below.

Let $p_\theta(\theta)$ be the probability density of a parameter θ and suppose a transformation $w = f(\theta)$, where w and θ has the same number of components. If $p_\theta(\theta)$ is a discrete distribution, and f is a one-to-one function, the density of w is given by

$$p_w(w) = p_\theta(f^{-1}(w)). \tag{6.15}$$

In the case of continues probability density p_θ and one-to-one transformation function,

the joint density of the transformed vector is

$$p_w(w) = |J| p_\theta(f^{-1}(w)), \tag{6.16}$$

where J is the Jacobian matrix of the transformation $\theta = f^{-1}(w)$ as a function of w. The Jacobian is a square matrix with entries (i, j) given by partial derivatives $\partial\theta_i/\partial w_j$.

As discussed in the context of Bayesian approach and HMC sampling many of the computation are done with the energy function (3.7) which is obtained for the transformed parameter as

$$\begin{aligned} E_w(w) &= -\log|J| - \log\Big(p(D \mid f^{-1}(w))\Big) - \log\Big(p_\theta(f^{-1}(w))\Big) \\ &= E_\theta(\theta) - \log|J|. \end{aligned} \tag{6.17}$$

In the HMC sampling also the gradients of the energy function E_w with respect to the parameters w are needed. A general expression for them is obtained as following

$$\begin{aligned} \frac{\partial E_w(w)}{\partial w} &= \frac{\partial}{\partial\theta}\left[E_\theta(\theta) - \log(|J|)\right]\frac{\partial\theta}{\partial w} \\ &= \left[\frac{\partial E_\theta(\theta)}{\partial\theta} - \frac{1}{|J|}\frac{\partial|J|}{\partial\theta}\right]\frac{\partial\theta}{\partial w}. \end{aligned} \tag{6.18}$$

6.3.2 Non-isotropic distribution

In a multivariate normal distribution, the eigenvalues and the eigenvectors of the covariance matrix can be used to study the properties of the distribution. Figure 6.2 illustrates how the eigenvalues of the covariance matrix affect on the shape of the distribution making it look like a cigar in a direction defined by the eigenvectors.

The sampling of latent values is conducted by HMC algorithm in which the new proposal state is generated by the dynamical simulations and it is accepted by the Metropolis rule. The dynamical simulations reduces the random walk behavior especially if the latent values have different size of variance and, thus, helps the algorithm move faster from other end of the distribution to the other. Hybrid Monte Carlo faces also its limitations if the

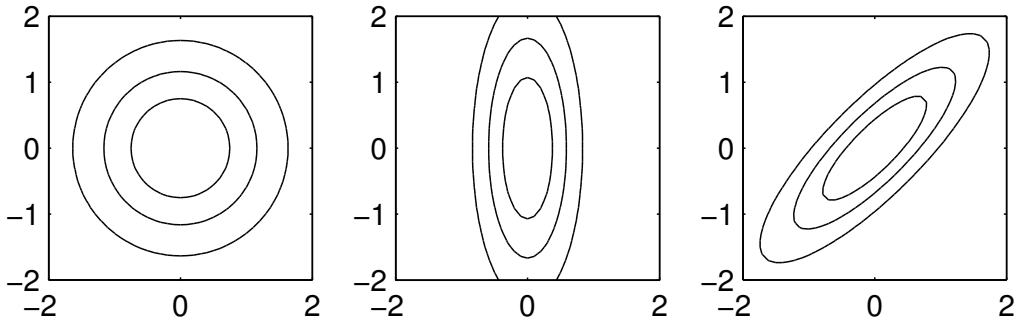


Figure 6.2: **Two dimensional normal distributions**. The distribution *on the left* represents isotropic normal distribution and covariance matrix with eigenvalues $\lambda_1 = \lambda_2 = 1$ and eigenvectors $v_1 = [0, 1]$, $v_2 = [1, 0]$. *On the middle* a normal distribution in which the variance for parameters differ from each others. The covariance matrix has eigenvalues $\lambda_1 = 0.5$ and $\lambda_2 = 4.0$ and eigenvectors $v_1 = [0, 1]$, $v_2 = [1, 0]$. *On the right* a normal distribution with correlating parameters. The covariance matrix has eigenvalues $\lambda_1 = \lambda_2 = 5.5$ and eigenvectors $v_1 = [-0.71, -0.71]$, $v_2 = [-0.71, 0.71]$.

variance differences and the correlation between sampled parameters are too large, reflecting to a highly cigar like distribution. In these situations it might be helpful to scale and rotate the parameter space into another and conduct the sampling in the resulting new parameter space. In the figure 6.2 the distributions on the left and on the right represent the easiest and the most troublesome ones to sample from. In the case of the distribution on the right an ideal transformation would transform the distribution similar to the one on the left. An overview of scaling for different algorithms is given, for example, by Roberts and Rosenthal (2001).

6.3.3 Approximate posterior variance

In order to reduce the inhomogeneity of the latent values $\mathbf{f}$ Christensen et al. (2006) have suggested to transform them with their approximate posterior covariance matrix Σ and to conduct the sampling in the resulting $\tilde{\mathbf{f}} = \Sigma^{-1/2}\mathbf{f}$ space. In this work the approach is followed for the full GP and extended for FITC sparse approximation.

The posterior distribution of latent values is a product of the normal prior and the Poisson likelihood. The likelihood is given a normal approximation in its mode and its precision

is approximated with a second derivative of the log Poisson in the mode

$$\Sigma_{\mathrm{l}}^{-1} \approx -\frac{\partial^2 \log(\mathrm{Poisson}(E\mu))}{\partial f^2} = E\mu. \tag{6.19}$$

The precision of the likelihood is thus a product of the age adjusted and the relative mortality risk. The approximate posterior precision is obtained as a sum of the precisions of the prior $\mathbf{K}^{-1}$ and the likelihood

$$\Sigma^{-1} = \mathbf{K}^{-1} + \mathrm{diag}\left[E_1\mu_1, ..., E_n\mu_n\right]. \tag{6.20}$$

In order to retain the reversibility of MCMC sampling the transformation may not depend on the sampled parameter, and thus μ is approximated with its prior mean 1. This should be a reasonably good approximation since μ's posterior variance is usually moderate in spatial epidemiology. The above equation leads to the observation that if the prior covariance is kept unchanged a large expected mortality rate leads to a smaller posterior variance than a small one. As well, because the number of deaths is modeled by Poisson distribution with mean $\mathbb{E}\left[Y_i\right] = E_i\mu_i$, a large numbers of death cases tend to be more informative about their mean than small ones. In the case of very rare diseases, with small number of death cases, the above characteristic results into a covariance matrix with a large number of very small eigenvalues and a few large ones and thus into a very narrow, high dimensional, cigar like distribution.

6.3.4 Transformation in FITC

In the FITC approximation $\mathbf{Q}_{\mathrm{f,f}} + \Lambda$, replaces the prior covariance $\mathbf{K}$, and the posterior precision transforms to

$$\Sigma_{\mathrm{FITC}}^{-1} = \left(\mathbf{Q}_{\mathrm{f,f}} + \Lambda\right)^{-1} + \mathrm{diag}\left[E_1\mu_1, ..., E_n\mu_n\right], \tag{6.21}$$

where Λ is a diagonal $n \times n$ matrix and $\mathbf{Q}_{\mathrm{f,f}}$ an $n \times n$ matrix of rank m. The transformation $\tilde{\mathbf{f}} = \Sigma_{\mathrm{FITC}}^{-1/2}\mathbf{f}$ could be done by evaluating the full matrix $\Sigma_{\mathrm{FITC}}^{-1}$ and taking a matrix square root of it. However, in this case the advantage of sparse approximation would be lost.

In order to extend the transformation to FITC approximation in a way the evaluation of the full covariance matrix is avoided a matrix inversion lemma (4.23) is first used to break up $\left(\mathbf{Q}_{\mathrm{f,f}} + \Lambda\right)^{-1}$ into computationally more efficient form, after which the posterior precision is obtained a relation

$$\Sigma_{\mathrm{FITC}}^{-1} = \Lambda^{-1} - \Lambda^{-1}\mathbf{K}_{\mathrm{f,u}}\left(\mathbf{K}_{\mathrm{u,u}} + \mathbf{K}_{\mathrm{u,f}}\Lambda^{-1}\mathbf{K}_{\mathrm{f,u}}\right)^{-1}\mathbf{K}_{\mathrm{u,f}}\Lambda^{-1} + \Sigma_{\mathrm{l}}^{-1}. \tag{6.22}$$

Next, denoting

$$\widehat{\Lambda}^{-1} = \Sigma_{l}^{-1} + \Lambda^{-1} \tag{6.23}$$

$$\mathbf{L} = \Lambda^{-1}\mathbf{K}_{\mathrm{f,u}}\mathrm{chol}\left[\mathbf{K}_{\mathrm{u,u}} + \mathbf{K}_{\mathrm{u,f}}\Lambda^{-1}\mathbf{K}_{\mathrm{f,u}}\right]^{-1} \tag{6.24}$$

and using the fact that $\mathbf{K}_{\mathrm{f,u}}^{\mathrm{T}} = \mathbf{K}_{\mathrm{u,f}}$ and $\left(\Lambda^{-1}\right)^{\mathrm{T}} = \Lambda^{-1}$ the posterior precision can be simplified into

$$\Sigma_{\mathrm{FITC}}^{-1} = \widehat{\Lambda}^{-1} - \mathbf{L}\mathbf{L}^{\mathrm{T}}, \tag{6.25}$$

where $\mathbf{L}$ is a $n \times m$ matrix and $\widehat{\Lambda}$ a diagonal $n \times n$ matrix.

So far only the notation of the posterior precision has been modified and no transformation of any kind have occurred. The first transformation needed is to scale the posterior precision so that the diagonal elements of $\widehat{\Lambda}$ are scaled to a constant, that is $\widehat{\Lambda} = \lambda\mathbf{I}$. To do this $\Sigma_{\mathrm{FITC}}^{-1}$ is multiplied by $\widehat{\Lambda}^{1/2}$ from left and right, which corresponds to transforming the latent values into a $\hat{\mathbf{f}} = \widehat{\Lambda}^{-1/2}\mathbf{f}$ space with approximate posterior precision

$$\widehat{\Sigma}_{\mathrm{FITC}}^{-1} = \mathbf{I} - \widehat{\Lambda}^{1/2}\mathbf{L}\mathbf{L}^{\mathrm{T}}\widehat{\Lambda}^{1/2}. \tag{6.26}$$

Here the matrix $\widehat{\Sigma}_{\mathrm{FITC}}^{-1}$ is of rank n and the matrix $\widehat{\Lambda}^{1/2}\mathbf{L}\mathbf{L}^{\mathrm{T}}\widehat{\Lambda}^{1/2}$ of rank m. The transformation to be done in the following, results in a scaling in the direction of the m largest eigenvalues instead of in the direction of all the eigenvalues as is the case with full GP.

It has been shown (e.g. Harville, 1997, theorem 21.9.1) that corresponding to any $n \times n$ symmetric non-negative definite matrix $\mathbf{A}$ there exists symmetric non-negative definite

matrix $\mathbf{R}$ such that $\mathbf{A} = \mathbf{R}^2$ and

$$\mathbf{R} = \mathbf{U}\text{diag}\left[\sqrt{d_1}, ..., \sqrt{d_n}\right]\mathbf{U}^{\mathrm{T}} = \mathbf{U}\mathbf{D}^{1/2}\mathbf{U}^{\mathrm{T}}, \tag{6.27}$$

where $d_1, ..., d_n$ are the eigenvalues of $\mathbf{A}$ and $\mathbf{U}$ is any $n \times n$ matrix such that $\mathbf{U}^{\mathrm{T}}\mathbf{A}\mathbf{U} = \mathbf{D}$ (that is for example a matrix of eigenvectors). It also holds that if $\mathbf{A}$ is a $n \times n$ matrix and k an arbitrary scalar, the eigenvectors and eigenvalues of the difference $\mathbf{A} - k\mathbf{I}$ are related to those of $\mathbf{A}$ itself in relatively simple way. If λ is the eigenvalue of $\mathbf{A}$ and $\mathbf{x}$ the eigenvector corresponding to it, the eigenvalue and eigenvector of $\mathbf{A} - k\mathbf{I}$ are $\lambda - k$ and $\mathbf{x}$ respectively (Harville, 1997, section 21.10).

Using the above results the m largest eigenvalues of $\widehat{\Sigma}_{\text{FITC}}^{-1}$ and the eigenvectors corresponding to them can be constructed by relations

$$\mathbf{U}\mathbf{S}\mathbf{U}^{\mathrm{T}} = \widehat{\Lambda}^{1/2}\mathbf{L}\mathbf{L}^{\mathrm{T}}\widehat{\Lambda}^{1/2} \tag{6.28}$$

$$\mathbf{D}^2 = \text{diag}\left[1 - \mathbf{S}_{11}, ..., 1 - \mathbf{S}_{mm}\right], \tag{6.29}$$

where $\mathbf{D}$ is an $m \times m$ diagonal matrix of square roots of the m largest eigenvalues to be used later and $\mathbf{U}$ an $n \times m$ matrix with the corresponding eigenvectors on its columns. The singular value decomposition $\mathbf{U}\mathbf{S}\mathbf{U}^{\mathrm{T}}$ can be found without explicitly forming the full $n \times n$ matrix by first defining a help matrix $\mathbf{B} = \mathbf{U}\mathbf{S}^{1/2}\mathbf{V}^{\mathrm{T}}$ and finding the eigenvalue decomposition of the $m \times m$ matrix

$$\mathbf{B}^{\mathrm{T}}\mathbf{B} = \mathbf{V}\mathbf{S}\mathbf{V}^{\mathrm{T}}, \tag{6.30}$$

after which the matrix of eigenvectors $\mathbf{U}$ is obtained from relation $\mathbf{U} = \mathbf{B}\mathbf{V}\mathbf{S}^{-1/2}$.

After solving $\mathbf{U}$, $\widehat{\Lambda}$ and $\mathbf{D}$ the transformation equations into a transformed space and back to the latent value space for FITC are obtained, respectively, from the following equations

$$\tilde{\mathbf{f}} = \widehat{\Lambda}^{-1/2}\,\mathbf{f} + \mathbf{U}\mathbf{D}\mathbf{U}^{\mathrm{T}}\widehat{\Lambda}^{-1/2}\,\mathbf{f} - \mathbf{U}\mathbf{U}^{\mathrm{T}}\widehat{\Lambda}^{-1/2}\,\mathbf{f} \tag{6.31}$$

$$\mathbf{f} = \widehat{\Lambda}^{1/2}(\tilde{\mathbf{f}} + \mathbf{U}\mathbf{D}^{-1}\mathbf{U}^{\mathrm{T}}\tilde{\mathbf{f}} - \mathbf{U}\mathbf{U}^{\mathrm{T}}\tilde{\mathbf{f}}). \tag{6.32}$$

The transformation done here is actually a combination of three steps. First the latent values are scaled by $\widehat{\Lambda}^{-1/2}$. After this the components of f along the dimensions defined by eigenvectors are first removed and then added with scaling by the square root of eigenvalues. For the transformation back to the latent values the steps are taken vice verse.

Algorithm 1 Transformation and re-transformation of latent values with their approximate posterior covariance. NOTE 1! The vectors of diagonal elements are represented with the same symbol as the diagonal matrices in the equations in the text. NOTE 2! Some of the notations are from Matlab. They are: ./ (pointwise division), .* (pointwise multiplication)

Input: $\mathbf{f}$ (latent values $\log(\mu)$), $\mathbf{E}$ (expected number of deaths), $\mathbf{K}_{\mathrm{f,u}}$, $\mathbf{K}_{\mathrm{u,u}}$, $\mathbf{k} = \left[\mathbf{K}_{\mathrm{f,f}}(1,1), ..., \mathbf{K}_{\mathrm{f,f}}(n,n)\right]$ (vector of diagonal elements of $\mathbf{K}_{\mathrm{f,f}}$)

if transform from $\mathbf{f}$ to $\tilde{\mathbf{f}}$ **then**
  $\mathbf{q} \leftarrow$ diagonals of $\mathbf{Q}_{\mathrm{f,f}}$ from chol$(\mathbf{K}_{\mathrm{u,u}}) \setminus \mathbf{K}_{\mathrm{f,u}}{}^{\mathrm{T}}$ (see eq. (6.3))
  $\Lambda \leftarrow k - \mathbf{q}$ (vector diag[Λ] of length n)
  $\widehat{\Lambda}^{-1} \leftarrow$ **E**.*exp(**f**)+1./Λ; (vector diag$\left[\widehat{\Lambda}\right]$ of length n, eq. (6.23))
  $\mathbf{K} \leftarrow \Lambda^{-1}\,\mathbf{K}_{\mathrm{f,u}}$ (evaluate as discussed in section 6.1.2)
  $\mathbf{L} \leftarrow \mathbf{K}\left(\left(\mathrm{chol}\left[\mathbf{K}_{\mathrm{u,u}} + \mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}\right]\right)^{-1}\right)^{\mathrm{T}}$ (This is faster and numerically more stable than $\mathbf{K}\mathrm{chol}\left[\left(\mathbf{K}_{\mathrm{u,u}} + \mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}\right)^{-1}\right]$)
  $\mathbf{B} \leftarrow \mathbf{L}*\widehat{\Lambda}^{1/2}$
  $\mathbf{S} \leftarrow$ eigenvalues of $\mathbf{B}^{\mathrm{T}}\mathbf{B}$ (a vector of length m eq. (6.30))
  $\mathbf{V} \leftarrow$ eigenvectors of $\mathbf{B}^{\mathrm{T}}\mathbf{B}$ ($m \times m$ matrix eq. (6.30))
  $\mathbf{U} \leftarrow \mathbf{BV}/\mathbf{S}^{\mathbf{1/2}}$
  $\mathbf{D} \leftarrow (1 - \mathbf{S})^{1/2}$ (this is a vector and thus the square root can be evaluated pointwise)
  save $\mathbf{D}$, $\mathbf{U}$ and $\widehat{\Lambda}$ (for use in re-transformation)
  $\hat{\mathbf{f}} \leftarrow \widehat{\Lambda}^{-1/2}\,\mathbf{f}$
  $\tilde{\mathbf{f}} \leftarrow \hat{\mathbf{f}} + \mathbf{U}\left[\left(\mathbf{D}\mathbf{U}^{\mathrm{T}} - \mathbf{U}^{\mathrm{T}}\right)\hat{\mathbf{f}}\right]$
  return $\tilde{\mathbf{f}}$
end if

if transform from $\tilde{\mathbf{f}}$ to $\mathbf{f}$ **then**
  load $\mathbf{D}$, $\mathbf{U}$ and $\widehat{\Lambda}$
  $\mathbf{f} \leftarrow \widehat{\Lambda}^{1/2}\left[\tilde{\mathbf{f}} + \mathbf{U}\left((\mathbf{D}^{-1}\mathbf{U}^{\mathrm{T}} - \mathbf{U}^{\mathrm{T}})\tilde{\mathbf{f}}\right)\right]$
  return $\mathbf{f}$
end if

6.4 Gradients of an energy function

6.4.1 Gradients with respect to hyperparameters

The hyperparameters and the latent values are sampled with hybrid Monte Carlo method, which needs the information about the derivatives of an energy function E with respect to the sampled parameters. In regression problem with full GP the gradients with respect to hyperparameters θ are obtained from

$$\begin{aligned}\frac{\partial E}{\partial \theta} &= \frac{\partial}{\partial \theta}\log(p(\mathbf{y}\,|\mathbf{x},\theta)) + \frac{\partial}{\partial \theta}\log(p(\theta|\gamma)) \\ &= \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_{\mathrm{f,f}}^{-1}\frac{\partial\,\mathbf{K}_{\mathrm{f,f}}}{\partial\theta}\right) - \frac{1}{2}\,\mathbf{y}^{\mathrm{T}}\,\mathbf{K}_{\mathrm{f,f}}^{-1}\frac{\partial\,\mathbf{K}_{\mathrm{f,f}}}{\partial\theta}\mathbf{K}_{\mathrm{f,f}}^{-1}\,\mathbf{y} + \frac{\partial}{\partial\theta}\log(p(\theta|\gamma)),\end{aligned} \tag{6.33}$$

where $\log(p(\theta|\gamma)$ is the term resulting from the hyperprior. If the Gaussian process is used with an arbitrary likelihood the energy function is obtained from the equation (4.16) as discussed in section 4.2.2. In that case the gradients with respect to θ are obtained from the above relation by changing the vector of training targets $\mathbf{y}$ to the latent value vector $\mathbf{f}$. In the energy function with an arbitrary likelihood only the covariance matrix $\mathbf{K}_{\mathrm{f,f}}$ and the prior $p(\theta)$ are functions of θ and thus the derivative of the likelihood $p(\mathbf{y}\,|g(\mathbf{f}))$ in equation (4.16) equals to zero. Below the discussion is given by considering the gradients in regression problem, but as mentioned, the results can also be ablied for other likelihoods by changing $\mathbf{y}$ in the equations into $\mathbf{f}$. The hyperprior term is also neglected in the following treatment in order to shorten the notation.

In the FITC approximation the covariance matrix $\mathbf{K}_{\mathrm{f,f}}$ is replaced by $\mathbf{Q}_{\mathrm{f,f}} + \Lambda$ and the gradient of an energy function is obtained from

$$\begin{aligned}\frac{\partial E(\theta)}{\partial\theta} =& \mathrm{tr}\left(\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}\frac{\partial(\mathbf{Q}_{\mathrm{f,f}}+\Lambda)}{\partial\theta}\right) \\ &- \frac{1}{2}\,\mathbf{y}^{\mathrm{T}}\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}\frac{\partial(\mathbf{Q}_{\mathrm{f,f}}+\Lambda)}{\partial\theta}\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}\mathbf{y}\,.\end{aligned} \tag{6.34}$$

In the case of full GP the entries $\frac{\partial\,\mathbf{K}_{\mathrm{f,f}}(i,j)}{\partial\theta}$ can be obtained directly from the derivatives of the covariance function $\frac{\partial k(\mathbf{x}_i,\mathbf{x}_j)}{\partial\theta}$, but with the FITC approximation the problem is some-

what more awkward, since

$$\frac{\partial(\mathbf{Q}_{\mathrm{f,f}}+\Lambda)}{\partial\theta} = \frac{\partial(\mathbf{Q}_{\mathrm{f,f}})}{\partial\theta} + \frac{\partial}{\partial\theta}\mathrm{diag}\left[\mathbf{K}_{\mathrm{f,f}}\right] - \frac{\partial}{\partial\theta}\mathrm{diag}\left[\mathbf{Q}_{\mathrm{f,f}}\right] \tag{6.35}$$

where the gradients of $\mathbf{Q}_{\mathrm{f,f}} = \mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{u,u}^{-1}\,\mathbf{K}_{\mathrm{u,f}}$ can not be evaluated without matrix operations. Snelson and Ghahramani (2006) had used a gradient ascent method for optimizing the parameters of GP with FITC sparse approximation in their work. However, the gradients in (6.34) could not be found in the literature and thus the evaluation and implementation of them played an essential role in this work. The derivation of gradients is conducted below.

There are two terms in (6.34) that are evaluated separately, the upper and the lower line of the equation, and they will be denoted as following

$$T = \mathrm{tr}\left(\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}\frac{\partial(\mathbf{Q}_{\mathrm{f,f}}+\Lambda)}{\partial\theta}\right) \tag{6.36}$$

$$V = \frac{1}{2}\,\mathbf{y}^{\mathrm{T}}\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}\frac{\partial(\mathbf{Q}_{\mathrm{f,f}}+\Lambda)}{\partial\theta}\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}\mathbf{y}\,. \tag{6.37}$$

The evaluation of both of the terms requires the expression of the gradients of $\mathbf{Q}_{\mathrm{f,f}}$, which can be evaluated straightforwardly into the form of

$$\frac{\partial\,\mathbf{Q}_{\mathrm{f,f}}}{\partial\theta} = \left[2\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{f,u}}\right] + \mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{u,u}}\right]\right]\left(\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\right)^{\mathrm{T}}. \tag{6.38}$$

This is an $n \times n$ matrix and thus it is not evaluated explicitly. First, to start with the evaluation of the term V a length n vector $\mathbf{b} = \mathbf{y}^{\mathrm{T}}\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}$ is constructed. This can be done efficiently by denoting $\mathbf{L} = \Lambda^{-1}\mathbf{K}_{\mathrm{f,u}}\mathrm{chol}\left[\mathbf{K}_{\mathrm{u,u}} + \mathbf{K}_{\mathrm{u,f}}\Lambda^{-1}\mathbf{K}_{\mathrm{f,u}}\right]^{-1}$, after which the vector is obtained from

$$\begin{aligned}\mathbf{b} &= \mathbf{y}^{\mathrm{T}}\left(\Lambda^{-1} + \Lambda^{-1}\,\mathbf{K}_{\mathrm{f,u}}\left(\mathbf{K}_{\mathrm{u,u}} + \mathbf{K}_{\mathrm{u,f}}\,\Lambda^{-1}\,\mathbf{K}_{\mathrm{f,u}}\right)^{-1}\mathbf{K}_{\mathrm{u,f}}\,\Lambda^{-1}\right)\\ &= \mathbf{y}^{\mathrm{T}}\,\Lambda^{-1} + \left(\mathbf{y}^{\mathrm{T}}\,\mathbf{L}\right)\left(\mathbf{y}^{\mathrm{T}}\,\mathbf{L}\right)^{\mathrm{T}}.\end{aligned} \tag{6.39}$$

Here the multiplications with diagonal matrix (or a vector of diagonal elements) are conducted as described earlier in the section 6.1.2. Now, by plugging in the gradients of $\mathbf{Q}_{\mathrm{f,f}}$

from (6.38) V can be expressed as

$$\begin{aligned} V =& \mathbf{b}\,\frac{\partial\,\mathbf{Q}_{\mathrm{f,f}}}{\partial\theta}\,\mathbf{b}^{\mathrm{T}} + \mathbf{b}\,\frac{\partial\,\Lambda}{\partial\theta}\,\mathbf{b}^{\mathrm{T}} \\ =& \left[\mathbf{b}\,2\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{f,u}}\right] + \mathbf{b}\,\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{u,u}}\right]\right]\left(\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\right)^{\mathrm{T}}\mathbf{b}^{\mathrm{T}} \\ &+ \mathbf{b}\,\frac{\partial}{\partial\theta}\left[\mathrm{diag}\left[\mathbf{K}_{\mathrm{f,f}}\right]\right]\mathbf{b}^{\mathrm{T}} - \mathbf{b}\,\frac{\partial}{\partial\theta}\left[\mathrm{diag}\left[\mathbf{Q}_{\mathrm{f,f}}\right]\right]\mathbf{b}^{\mathrm{T}}, \end{aligned} \tag{6.40}$$

where the first and the second term can be evaluated without forming an $n \times n$ matrix if the calculation are conducted in right order. In order to proceed with the third term a diagonal matrix $\mathbf{B} = \mathrm{diag}\left[b_1^2, b_2^2, ..., b_n^2\right]$ is defined so that its diagonal elements are the elements of $\mathbf{b}$ squared, and thus the third term can be modified into

$$\begin{aligned} \mathbf{b}\,\frac{\partial(\mathrm{diag}\left[\mathbf{Q}_{\mathrm{f,f}}\right])}{\partial\theta}\,\mathbf{b}^{\mathrm{T}} =& \mathrm{tr}\left(\mathbf{b}\,\frac{\partial(\mathrm{diag}\left[\mathbf{Q}_{\mathrm{f,f}}\right])}{\partial\theta}\,\mathbf{b}^{\mathrm{T}}\right) \\ =& \mathrm{tr}\left(\mathbf{B}\frac{\partial(\mathrm{diag}\left[\mathbf{Q}_{\mathrm{f,f}}\right])}{\partial\theta}\right) \\ =& \mathrm{tr}\left(\mathbf{B}\frac{\partial(\mathbf{Q}_{\mathrm{f,f}})}{\partial\theta}\right), \end{aligned} \tag{6.41}$$

where the last step is taken by noticing that the multiplication by a diagonal matrix from left corresponds to multiplying the columns of $\mathbf{B}$ with the respective diagonal elements of $\frac{\partial}{\partial\theta}\mathrm{diag}\left[\mathbf{Q}_{\mathrm{f,f}}\right]$. The same result is obtained by multiplying the rows of $\frac{\partial}{\partial\theta}\,\mathbf{Q}_{\mathrm{f,f}}$ by the respective diagonals of $\mathbf{B}$ (see discussion in section 6.1.2), which was defined to be diagonal and thus the diagonal operator can be neglected. Now, by taking in the $\frac{\partial\,\mathbf{Q}_{\mathrm{f,f}}}{\partial\theta}$ and using the fact that $\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$, where $\mathbf{B}$ is an $m \times n$ matrix and $\mathbf{A}$ an $n \times m$ matrix, this can be modified futrher as follows

$$\begin{aligned} \mathbf{b}\,\frac{\partial(\mathrm{diag}\left[\mathbf{Q}_{\mathrm{f,f}}\right])}{\partial\theta}\,\mathbf{b}^{\mathrm{T}} =& 2\mathrm{tr}\left(\left(\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\right)^{\mathrm{T}}\mathbf{B}\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{f,u}}\right]\right) - \\ & \mathrm{tr}\left(\left(\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\right)^{\mathrm{T}}\mathbf{B}\,\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{u,u}}\right]\right). \end{aligned} \tag{6.42}$$

Above, the expressions inside the trace operator form an $n \times n$ matrix, if the matrix multiplications are conducted and the trace is taken after that. However, this can be avoided by noticing that the trace of a matrix product between an $n \times m$ matrix $\mathbf{A}$ and an $m \times n$

matrix $\mathbf{C}$ can be written as

$$\mathrm{tr}(\mathbf{AC}) = \Sigma_{i=1}^{n} \Sigma_{j=1}^{m} a_{ij} c_{ji} \tag{6.43}$$

which is actually a dot product of vectors $\mathbf{a} = [a_{11}, a_{12}, ..., a_{1n}, a_{21}, ..., a_{2n}, ..., a_{mn}]$ and $\mathbf{c} = [c_{11}, c_{21}, ..., c_{n1}, c_{12}, ..., c_{n2}, ..., c_{nm}]$. The evaluation of the traces in (6.42) can thus be handled with a dot product of two $1 \times nm$ vectors. The needed results for above equalities are given, for example, by Harville (1997, pages 50–52). Furthermore, by writing the term $\left(\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\right)^{\mathrm{T}}\mathbf{b}^{\mathrm{T}}$ in (6.40) as $\left(\mathbf{b}\,\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\right)^{\mathrm{T}}$, the final solution of V can be obtained from

$$\begin{aligned} V = &\left[2\,\mathbf{b}\,\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{f,u}}\right] + \mathbf{b}\,\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{u,u}}\right]\right]\left(\mathbf{b}\,\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\right)^{\mathrm{T}} + \mathbf{b}\,\frac{\partial(\mathrm{diag}\left[\mathbf{K}_{\mathrm{f,f}}\right])}{\partial\theta}\,\mathbf{b}^{\mathrm{T}} \\ &- 2\mathrm{tr}\left(\left(\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\right)^{\mathrm{T}}\mathbf{B}\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{f,u}}\right]\right) + \mathrm{tr}\left(\left(\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\right)^{\mathrm{T}}\mathbf{B}\,\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{u,u}}\right]\right), \end{aligned} \tag{6.44}$$

which can be evaluated without forming any $n \times n$ matrix and enables the use of intermediate results in several places.

The evaluation of the term T is begun by partitioning it as following

$$\begin{aligned} T = &\mathrm{tr}\left(\left(\mathbf{Q}_{\mathrm{f,f}} + \Lambda\right)^{-1}\frac{\partial\,\mathbf{Q}_{\mathrm{f,f}}}{\partial\theta}\right) + \mathrm{tr}\left(\left(\mathbf{Q}_{\mathrm{f,f}} + \Lambda\right)^{-1}\frac{\partial}{\partial\theta}\mathrm{diag}\left[\mathbf{K}_{\mathrm{f,f}}\right]\right) - \\ &\mathrm{tr}\left(\left(\mathbf{Q}_{\mathrm{f,f}} + \Lambda\right)^{-1}\frac{\partial}{\partial\theta}\mathrm{diag}\left[\mathbf{Q}_{\mathrm{f,f}}\right]\right). \end{aligned} \tag{6.45}$$

where the matrix inversion lemma can be used for the first term. The second term can be evaluated by first solving $\mathrm{diag}\left[\left(\mathbf{Q}_{\mathrm{f,f}} + \Lambda\right)^{-1}\right]$, and then using the fact that $\mathrm{tr}\,(\mathbf{A}\mathrm{diag}\,[\mathbf{B}]) = \mathrm{tr}\,(\mathrm{diag}\,[\mathbf{A}]\,\mathrm{diag}\,[\mathbf{B}])$ The multiplication by a diagonal matrix from left in the last term corresponds to multiplying the columns of $\left(\mathbf{Q}_{\mathrm{f,f}} + \Lambda\right)^{-1}$ with the respective diagonal elements of $\frac{\partial}{\partial\theta}\mathrm{diag}\left[\mathbf{Q}_{\mathrm{f,f}}\right]$. The same result is obtained by multiplying the rows of $\frac{\partial}{\partial\theta}\,\mathbf{Q}_{\mathrm{f,f}}$ by the respective diagonals of $\left(\mathbf{Q}_{\mathrm{f,f}} + \Lambda\right)^{-1}$. Thus using the same idea as in (6.41) the last term can be changed into $\mathrm{tr}\left(\mathrm{diag}\left[\left(\mathbf{Q}_{\mathrm{f,f}} + \Lambda\right)^{-1}\right]\frac{\partial}{\partial\theta}\left[\mathbf{Q}_{\mathrm{f,f}}\right]\right)$. By plugging in the derivative of $\mathbf{Q}_{\mathrm{f,f}}$ from equation (6.38) and using the fact that $\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$ as above, the

expression can be modified into

$$
\begin{aligned}
T =& 2\mathrm{tr}\left(\left(\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\right)^{\mathrm{T}}\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{f,u}}\right]\right)+\\
&\mathrm{tr}\left(\left(\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\right)^{\mathrm{T}}\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{u,u}}\right]\right)+\\
&\mathrm{tr}\left(\mathrm{diag}\left[\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}\right]\frac{\partial}{\partial\theta}\left[\mathrm{diag}\left[\mathbf{K}_{\mathrm{f,f}}\right]\right]\right)-\\
&2\mathrm{tr}\left(\mathrm{diag}\left[\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}\right]\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{f,u}}\right]\left(\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\right)^{\mathrm{T}}\right)+\\
&\mathrm{tr}\left(\mathrm{diag}\left[\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}\right]\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{u,u}}\right]\left(\mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}\right)^{\mathrm{T}}\right). \qquad (6.46)
\end{aligned}
$$

Earlier it was mentioned that the evaluation of trace can be changed to a dot product of two vectors formed of the matrices. Thus by conducting the operations above in a right order, the calculation of T can be conducted without forming any $n \times n$ matrix.

Algorithm 2 Calculate the gradients of minus log likelihood. Note! Here the notation $\mathbf{C}(:)$ represents a vector $[c_{11}, c_{21}, ..., c_{n1}, c_{12}, ..., c_{n2}, ..., c_{nn}]^{\mathrm{T}}$.

Input: $\mathbf{K}_{\mathrm{f,u}}$, $\mathbf{K}_{\mathrm{u,u}}$, $\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{f,u}}\right]$, $\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{u,u}}\right]$ $\mathbf{y}$, $\mathbf{k} = \left[\mathbf{K}_{\mathrm{f,f}}(1,1), ..., \mathbf{K}_{\mathrm{f,f}}(n,n)\right]$

% First evaluate help matrices
$\mathbf{b} \leftarrow \mathbf{y}^{\mathrm{T}}\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}$ (evaluate as in (6.39))
$\mathbf{A} \leftarrow \mathbf{K}_{\mathrm{f,u}}\,\mathbf{K}_{\mathrm{u,u}}^{-1}$
$\mathbf{F} \leftarrow \mathbf{A}^{\mathrm{T}}\mathbf{B}$
$\mathbf{G} \leftarrow \mathbf{bA}$
$\mathbf{M} \leftarrow \mathbf{A}^{\mathrm{T}}\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}$
$\mathbf{q} \leftarrow \mathrm{diag}\left[\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}\right]$ ($1 \times n$ vector of diagonal elements)
$\mathbf{P} \leftarrow \mathbf{A}\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{u,u}}\right]$
$\mathbf{R} \leftarrow 2\mathrm{diag}\left[\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}\right]\frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{f,u}}\right]$
$\mathbf{W} \leftarrow \mathrm{diag}\left[\left(\mathbf{Q}_{\mathrm{f,f}}+\Lambda\right)^{-1}\right]\mathbf{P}$
% Then evaluate the gradient
$V \leftarrow 2 * \frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{f,u}}\right] + \mathbf{G} * \frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{u,u}}\right]$
$V \leftarrow V * \mathbf{G}^{\mathrm{T}} + (\mathbf{b}.*\mathbf{k}) * \mathbf{b}^{\mathrm{T}}$
$V \leftarrow V + 2 * \left(\mathbf{F}^{\mathrm{T}}(:)\right)^{\mathrm{T}} * \frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{f,u}}(:)\right] + \left(\mathbf{F}^{\mathrm{T}}(:)\right)^{\mathrm{T}} * \mathbf{P}(:)$
$T \leftarrow 2 * (\mathbf{M}^{\mathrm{T}}(:))^{\mathrm{T}} * \frac{\partial}{\partial\theta}\left[\mathbf{K}_{\mathrm{f,u}}(:)\right] + (\mathbf{M}^{\mathrm{T}}(:))^{\mathrm{T}} * \mathbf{P}(:) + q * k^{\mathrm{T}}$
$T \leftarrow T + \mathbf{R}^{\mathrm{T}}(:))^{\mathrm{T}} * \mathbf{A}^{\mathrm{T}}(:) + \mathbf{W}^{\mathrm{T}}(:))^{\mathrm{T}} * \mathbf{A}^{\mathrm{T}}(:)$
return $T + V$

6.4.2 Gradients with respect to latent values

In the regression problem the integration over latent values can be conducted analytically, as shown in the equation (4.10). However, this is not possible with an arbitrary likelihood, in which case the integral over latent values has to be approximated. In this work the approximation is conducted by sampling the latent values by hybrid Monte Carlo, which needs the gradients of an energy function with respect to the latent values

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{f}} &= \frac{\partial}{\partial \mathbf{f}}\left[-\log\left(p(\mathbf{y}|g(\mathbf{f}))\right) - p\left((\mathbf{f}|\theta)\right)\log\left(p(\theta)\right)\right] \\ &= \frac{\partial}{\partial \mathbf{f}}\left[-\log\left(p(\mathbf{y}|g(\mathbf{f}))\right)\right] - \frac{\partial}{\partial \mathbf{f}}\left[-\frac{1}{2}\,\mathbf{f}^{\mathrm{T}}\,\mathbf{K}_{\mathrm{f,f}}^{-1}\,\mathbf{f}\right], \\ &= \frac{\partial}{\partial \mathbf{f}}\left[-\log\left(p(\mathbf{y}|g(\mathbf{f}))\right)\right] - \mathbf{K}_{\mathrm{f,f}}^{-1}\,\mathbf{f}\,. \end{aligned} \tag{6.47}$$

Again, in the case of FITC approximation the covariance matrix $\mathbf{K}_{\mathrm{f,f}}$ is changed to $\mathbf{Q}_{\mathrm{f,f}} + \Lambda$ and the gradient of an energy function with respect to latent values is obtained from

$$\frac{\partial E}{\partial \mathbf{f}} = \frac{\partial}{\partial \mathbf{f}}\left[-\log p(\mathbf{y}|g(\mathbf{f}))\right] - \left(\mathbf{Q}_{\mathrm{f,f}} + \Lambda\right)^{-1}\mathbf{f}\,. \tag{6.48}$$

In the model used here the likelihood $p(\mathbf{y}|g(\mathbf{f}))$ is Poisson with mean $\exp(f)\mathbf{E}$, where $\mathbf{E}$ is the expected number of deaths. Thus the gradient term resulting from likelihood is

$$\frac{\partial}{\partial \mathbf{f}}\left[-\log\left(p(\mathbf{y}|g(\mathbf{f}))\right)\right] = \exp(\mathbf{f})\mathbf{E} - \mathbf{y}\,. \tag{6.49}$$

Here it can be concluded that the gradients of an energy function with respect to latent values are computationally lot faster than the gradients with respect to the hyperparameters.

Chapter 7

Results on case problems

The model and methods discussed above were tested with four sets of data. The sets of data consisted of the mortality due to two different diseases, *cerebral vascular diseases* and *alcohol-related diseases*, in the time interval 1995-1999. The data sets were studied with lattice resolutions of 20km × 20km and 10km × 10km, resulting in 915 and 3193 data points respectively.

In the following sections, the performance and the results of FITC sparse Gaussian process are compared to the performance and the results of a full Gaussian process. The discussion includes the treatment of the MCMC simulations and model comparison using DIC. Some of the results for the specific problems are illustrated by maps.

7.1 Examples of maps

The final products of the disease mapping analysis are the maps representing the spatial variations in the disease risk. When creating such a map, the map-maker has to decide what information he wants to present, and how, he is going to do it. Throughout the discussion here, the focus has been to define the relative risk in different areas, and thus the relative risk is a natural variable to present on a map. However, as discussed earlier

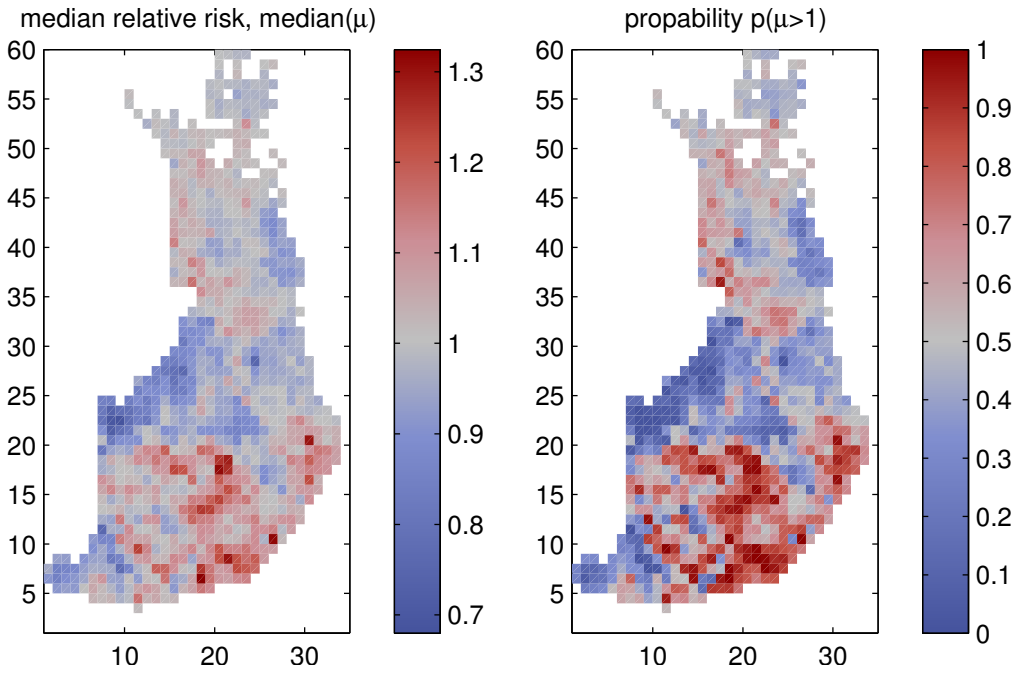
the results of Bayesian analysis are the probability distributions of variables of interest, and thus there is no single value for the relative risk to present, but there is a posterior knowledge of how probable certain values of the relative risk are in certain areas.

Presenting the whole distribution of the relative risk in all areas of the map is practically impossible and thus it has to be decided how to present the posterior information with single parameter values. Natural choices to plot are the mean or median of the relative risk, but they do not provide any information about the confidential intervals, when plotted alone. One possibility is to highlight areas where the relative risk is over certain value with a certain minimum probability. For example, areas where $\mu > 1.05$ with probability $p(\mu > 1.05) > 0.8$. However, in this approach the areas fulfilling the requirements can not be distinguished.

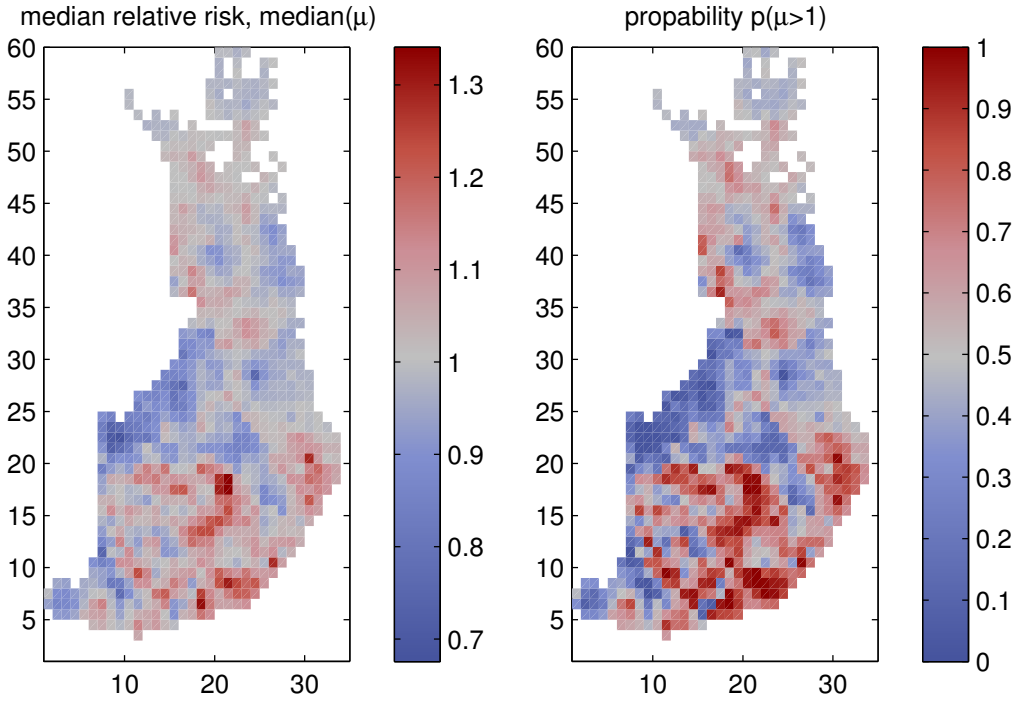
Here the posterior knowledge about relative risk is presented using two maps simultaneously. The other map presents the median of the relative risk and the other presents the probability of the relative risk being over 1, $p(\mu > 1|D)$. This was chosen for plotting, because the probability mass of the relative risk is distributed equally on both sides of the median of the risk. Thus it is equally probable that the relative risk is smaller or larger than the plotted value. The median map gives a crude estimate of the differences between relative risk levels in different areas, but other useful information is how probable it is that the relative risk is higher or lower to one. The map presenting $p(\mu > 1|D)$ represents the information of how probably the risk is increased or decreased in certain areas.

The maps presented here are the ones created by the best full GP and FITC models. With all the data sets these were the models with exponential covariance function. The models are compared in section 7.4.

The maps 7.1 and 7.2 present the results from the data aggregated into 20km$\times$20km lattice. The maps are drawn in a grid of size of 35$\times$60, where the side of a cell is in nature 20km. The resolution is rather rough, but areas of elevated and decreased disease risk can be distinguished from the maps. Both of the maps reveal areas, where the disease risk is elevated statistically significantly and especially the maps of alcohol related diseases reveal rather high relative risk in the eastern parts of the Finland.

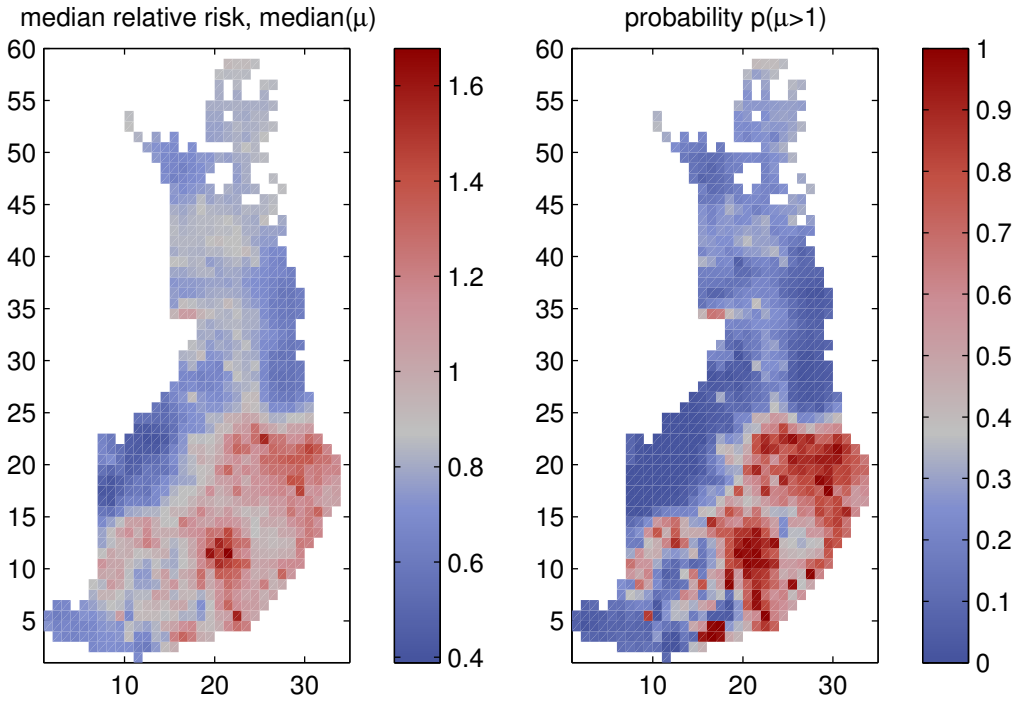


(a) FITC sparse approximation

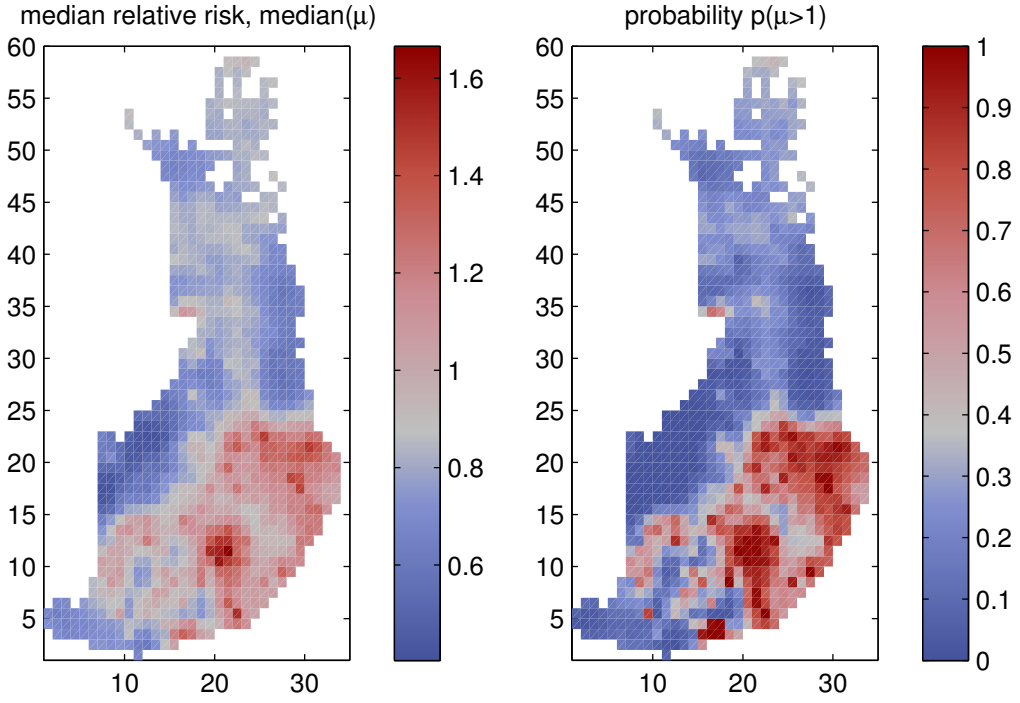


(b) Full Gaussian process

Figure 7.1: **The relative risk surface of the cerebral vascular diseases and the surface of $p(\mu > 1)$ in 20km×20km lattice.** The maps are results from the full GP and FITC approximation with exponential covariance function. These were the best full and sparse models in the case of the 20km×20km lattice. See table 7.3.



(a) FITC sparse approximation



(b) Full Gaussian process

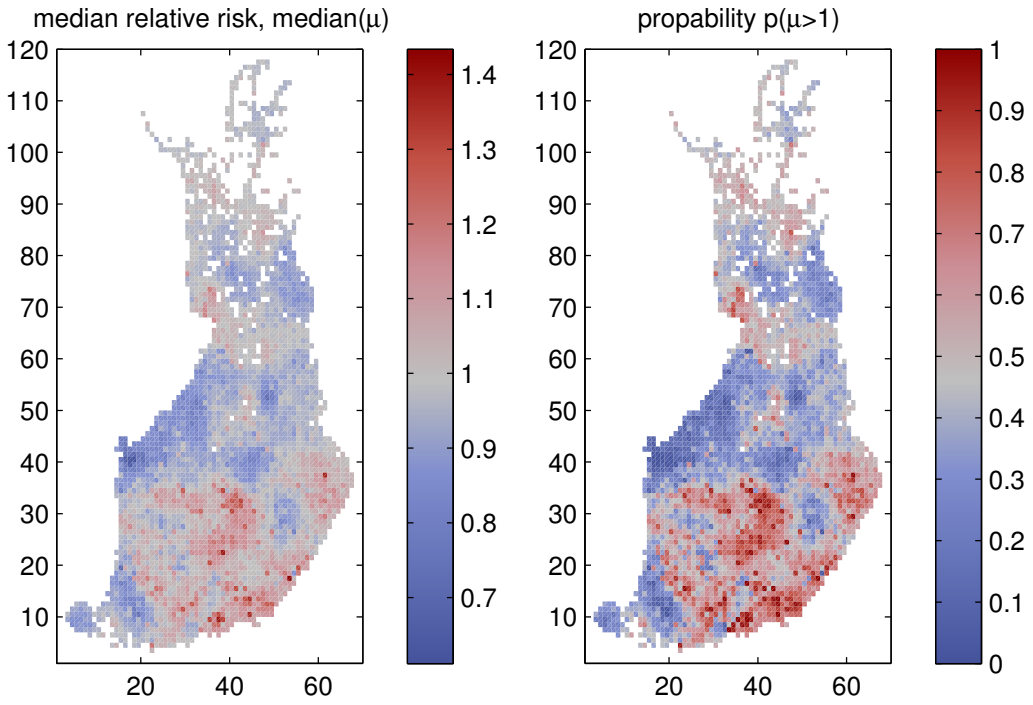
Figure 7.2: **The relative risk surface of the alcohol related diseases and the surface of $p(\mu > 1)$ in 20km×20km lattice.** The maps are results from the full GP and FITC approximation with exponential covariance function. These were the best full and sparse models in the case of the 20km×20km lattice. See table 7.4.

The maps on a 35×60 resolution are rather rough, but the models trained with the 20km$\times$ 20km lattice data, can be used to smooth the map presentation by making predictions in a denser grid. This may result in visually better appearance, but does not provide any extra information. The aggregation of data into 20km$\times$20km lattice has already smoothed the data and thus lost some of it spatial information. In the figures 7.3 and 7.4 the disease maps are presented in a 70×120 grid, representing a 10km$\times$10km lattice in nature. The other map in the figures is a prediction of full GP trained by the 20km$\times$20km lattice data into a denser grid, and the other is a product of an FITC approximation trained by the 10km$\times$10km lattice data.

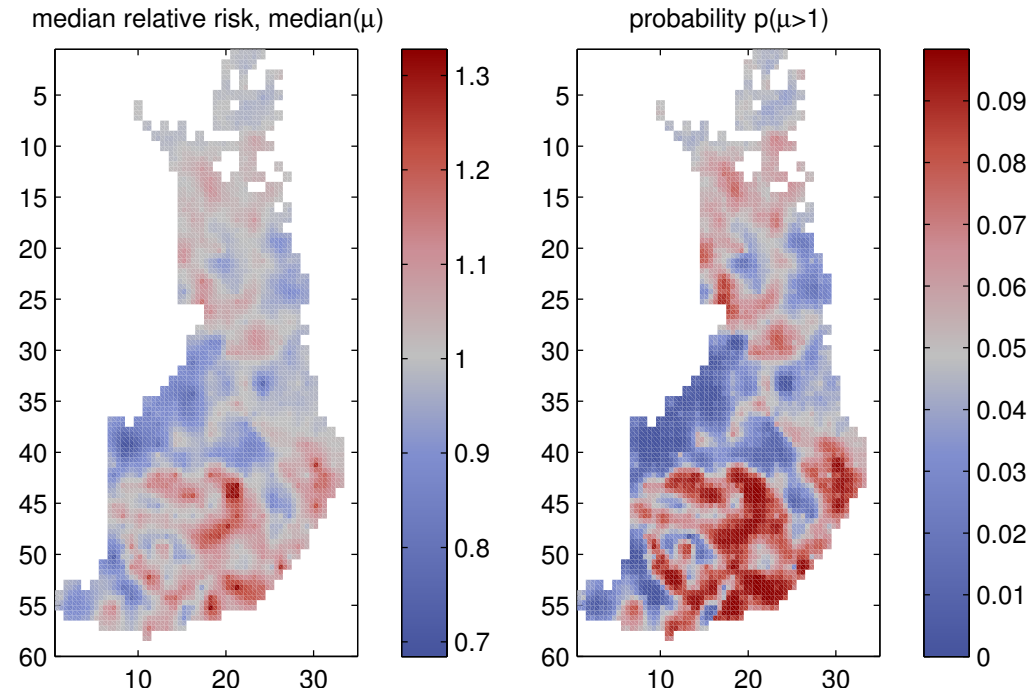
The maps in figures 7.3 and 7.4 look visually better than the ones plotted with smaller resolution. The main difference between the two maps in both figures is that the map of FITC approximation is more sharp-featured than the map of full Gaussian process model trained by the 20km$\times$20km lattice data. Making the predictions from a 35×60 grid to a denser 70×120 grid has smoothed out the sharp features and made the appearance of the map better. This kind of smoothing could be done also with the model trained by the 10km$\times$10km lattice data. The maps 7.3(a) and 7.4(a) reveal more peaks of elevated risk than the maps 7.4(b) and 7.3(b), which suggest that there may be rather small scale variations in the disease risk that were smoothed out already in aggregation of data into a 20km$\times$20km lattice. This explains also, why the exponential covariance function was found the best model for the 10km$\times$10km lattice data, as it is the least smooth of the covariance functions used (see section 7.4).

7.2 Sampling from the posterior

The data sets were rather different in nature as there were over three times more death cases in cerebral vascular diseases than in the alcohol related diseases. However, both data sets reflected diseases of rather small number of deaths, roughly 18 000 deaths in the cerebral vascular diseases and about 5200 in the alcohol-related diseases. As discussed in section 6.3.3, a small number of death cases lead to a posterior covariance matrix of relative risk with a large number of small eigenvalues and only a few large ones. This in

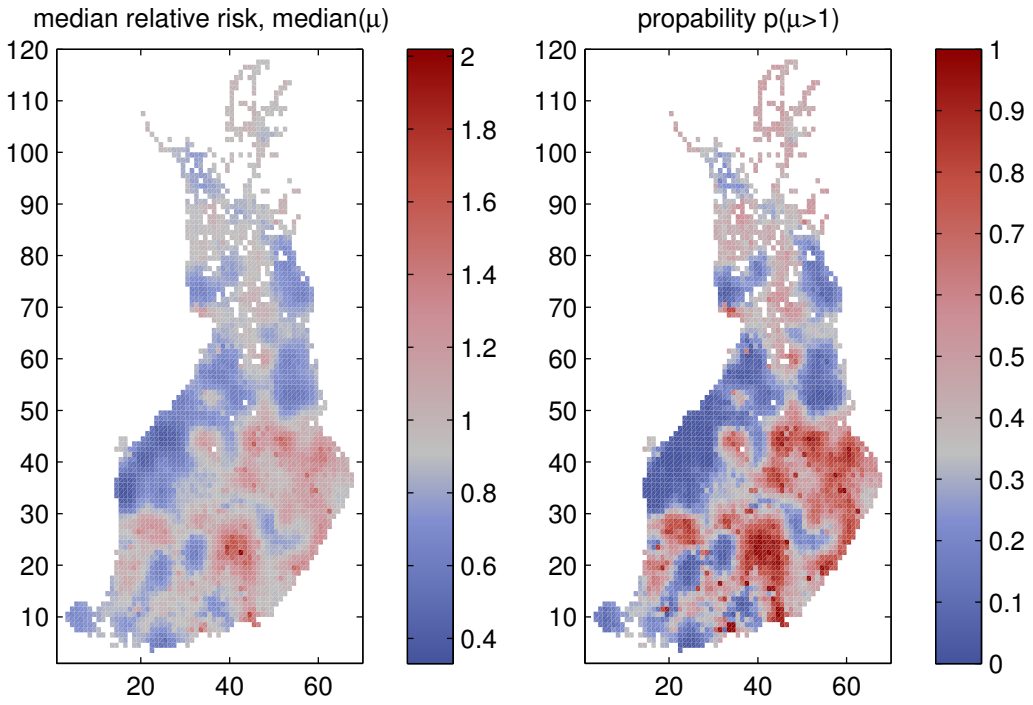


(a) FITC sparse approximation for 10km×10km lattice data

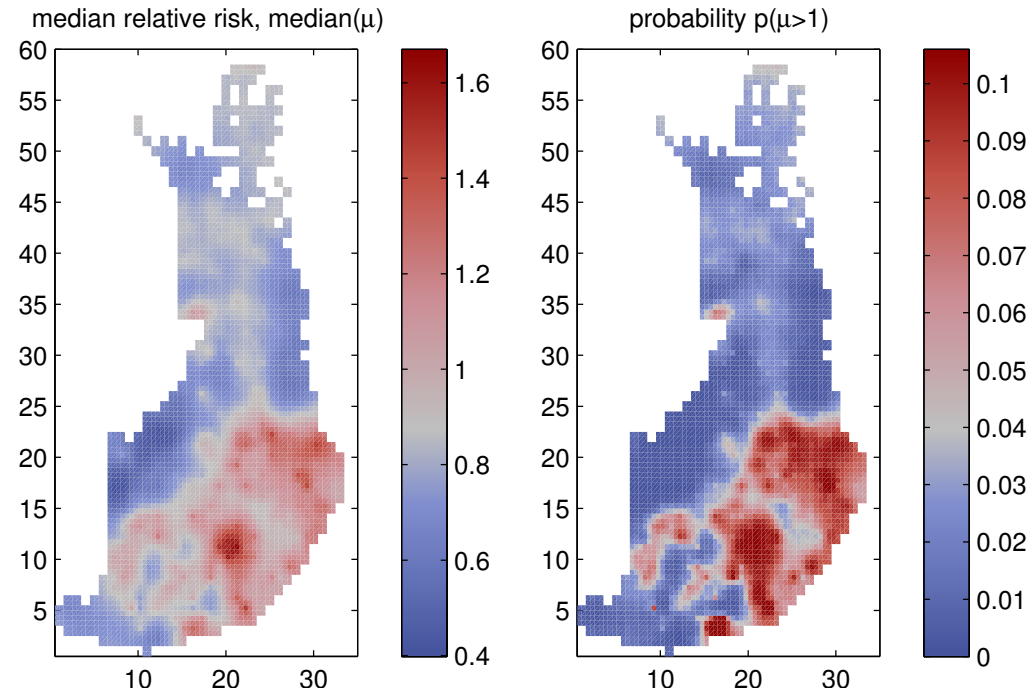


(b) Prediction of full GP trained with 20km×20km lattice data into 10km×10km lattice

Figure 7.3: **The relative risk surface of the cerebral vascular diseases and the surface of $p(\mu > 1)$ in 10km×10km lattice.** The maps are results from the FITC approximation for the 10km×10km lattice data and from the full GP trained by 20km×20km lattice data, which is used to make predictions in the denser grid. Both of the models have an exponential covariance function. These were the best sparse and predictive full models. See tables 7.5 and 7.3.



(a) FITC sparse approximation for 10km×10km lattice data



(b) Prediction of full GP trained with 20km×20km lattice data into 10km×10km lattice

Figure 7.4: **The relative risk surface of the alcohol related diseases and the surface of $p(\mu > 1)$ in 10km×10km lattice.** The maps are results from the FITC approximation for the 10km×10km lattice data and from the full GP trained by 20km×20km lattice data, which is used to make predictions in the denser grid. Both of the models have an exponential covariance function. These were the best sparse and predictive full models. See tables 7.5 and 7.4.

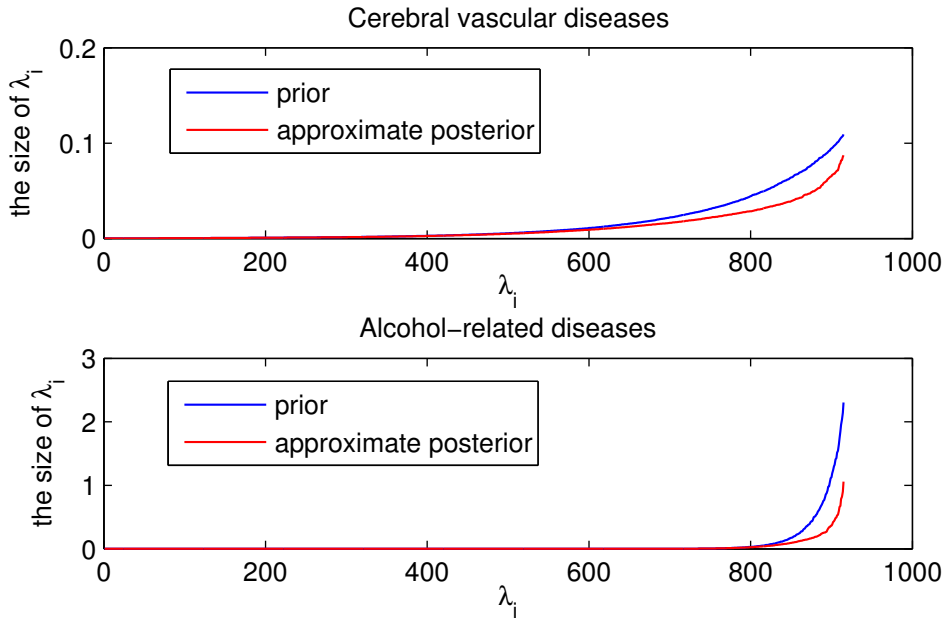


Figure 7.5: **The eigenvalues of prior and approximate posterior covariance matrix in the case of study data, plotted in the ascending order**. The covariance has been evaluated with posterior mean values of covariance function parameter. As discussed in section 6.3.3 a small number of death cases leads to a posterior covariance matrix with large number of small eigenvalues and only few large ones. This in turn represents a heavily non-isotropic and/or correlated joint posterior distribution for the relative risk, demonstrated in the figure 6.2. Here it can be concluded that the alcohol-related diseases data is more cigar like than the cerebral vascular diseases

turn may be a result of heavily non-isotropic and/or correlated joint posterior distribution demonstrated in the figure 6.2. The eigenvalues of the approximate posterior of the relative risk are shown in the figure 7.5, from which it is seen that the posterior with both data sets is rather cigar like.

In the case of cerebral vascular diseases data the parameters were significantly easier to sample than in the case of alcohol related diseases. The bottle neck of sampling with cerebral vascular diseases data were both, the length-scale and the magnitude and in the case of alcohol related diseases the length scale. The sampling with 915 data points and 221 inducing inputs was easier and the convergence faster than with 3193 data points and 238 inducing inputs. This may be a result of more data points and less inducing inputs per data points in the case of 10km$\times$10km lattice.

The sampling from joint posterior of hyperparameters $\theta = \left[l, \sigma_{\cdot}^2\right]$ and the latent values $\mathbf{f}$ is performed by Gibbs sampling, where the sampling from the conditional distributions $p(\theta|\,\mathbf{f}, D)$ and $p(\mathbf{f}\,|\theta, D)$ is conducted via hybrid Monte Carlo method. Hybrid Monte Carlo is efficient to sample from the conditional posteriors alone, but since the covariance function parameters, and especially the length-scale, are heavily dependent on the latent values the Gibbs sampling from the joint posterior $p(\theta, \mathbf{f}\,|D)$ is slow mixing. Large changes in the latent values result in significant changes in the posterior of θ, which in turn increases the random walk behavior of hyperparameter sampling, and thus leads to a high autocorrelation time. The latent value sampling is more robust for the changes in the covariance function parameters, since they are transformed by their approximate posterior. The change in the hyperparameters reflects to a change in the prior of latent values, which is taken into account also in the transformation (6.31). Christensen et al. (2006) suggested also a transformation for the hyperparameters, but it did not help with the models used here.

In this work, the mixing of hyperparameters was improved by using HMC with persistence, which reduces the random walk behavior. A good persistence parameter λ_{pers} value was around 0.9-0.95, which means that 90% of the original momentum is changed after 21 trajectories (see section 6.2.3). The mixing of the parameters can be altered by the other sampling options as well, and there are rather many options that can be tuned in the HMC method. To test the sampling of hyperparameters and latent values, and to find good sampling options, a number of chains were sampled for each data set and each model, and the best found were chosen for the final simulations. The options in the sampling were unique for all the models, and they were tuned to set the rejection rate around 0.04–0.1. The resulted autocorrelation times and sampling time for one sample are shown in the tables 7.1 and 7.2. The options used can be concluded as following:

1. Cerebral vascular diseases data.

   Hyperparameter sampling: Persistence parameter $\lambda_{\text{pers}} = 0.9$–$0.95$, stepsize $\delta t = 0.01$–0.015, number of leapfrog steps $L = 4$–5. Latent value sampling: No persistence, stepsize $\delta t = 0.15$–0.2, number of leapfrog steps $L = 10$.

2. Alcohol related diseases data.

Covariance function	full/ FITC	cerebral vascular diseases l	σ^2	$\log(\mu)$	alcohol related diseases l	σ^2	$\log(\mu)$
k_{sexp}	FITC	54	26	4	121	25	12
	full	129	34	9	131	8.3	11
k_{exp}	FITC	35	26	3	24	7.6	2.7
	full	26	14	3	23	22	4.0
$k_{\nu=3/2}$	FITC	27	14	3	55	12	5.1
	full	45	25	4	216	19	36
$k_{\nu=5/2}$	FITC	65	38	4	186	9.0	37
	full	65	42	6	268	1.9	40

Table 7.1: **The autocorrelation times for full and FITC sparse Gaussian process in the case of 20km × 20km lattice, 915 data points**. The autocorrelation time in the case of latent values $\log(\mu)$ is the time under which 97.5% of latent values are. With the sampling parameters used a CPU-time needed for one (dependent) sample in Intel Pentium 4 (1700MHz, 1GB memory) workstation for the FITC approximation was approximately 9s and for the full GP 19s.

Covariance function	full/ FITC	cerebral vascular diseases l	σ^2	$\log(\mu)$	alcohol related diseases l	σ^2	$\log(\mu)$
k_{sexp}	FITC	49	38	1.7	141	50	3.8
k_{exp}	FITC	25	26	1.5	22	7.8	1.8
$k_{\nu=3/2}$	FITC	43	75	2.7	435	156	19
$k_{\nu=5/2}$	FITC	44	52	2.6	610	120	71

Table 7.2: **The autocorrelation times for full and FITC sparse Gaussian process in the case of 10km × 10km lattice, 3193 data points**. The autocorrelation time in the case of latent values $\log(\mu)$ is the time under which 97.5% of latent values are. With the sampling parameters used the CPU-time needed for one (dependent) sample in Intel Pentium 4 (1700MHz, 1GB memory) workstation for the FITC approximation was approximately 54s.

Hyperparameter sampling: Persistence parameter $\lambda_{\text{pers}} = 0.9$–$0.95$, stepsize $\delta t = 0.01$–0.015, number of leapfrog steps $L = 4$–5, number of trajectories 3 (Mátern $\nu = 3/2$ only 1). Latent value sampling: No persistence, stepsize $\delta t = 0.015$–0.2, number of leapfrog steps $L = 10$.

7.3 Time needed for the sampling

The time needed for the posterior simulations with Gaussian processes is highly dependent on the size of the data, and in the case of FITC approximation also on the number of inducing inputs. As the sampling from the conditional distributions $p(\theta|\,\mathbf{f}, D)$ and $p(\mathbf{f}\,|\theta, D)$ is conducted via hybrid Monte Carlo method, the evaluation of gradients with respect to the hyperparameters and the latent values are required at each iteration round as many times as there are leapfrog steps. The gradients are derived in the section 6.4, and it can be concluded already from the equations (6.44), (6.46) and (6.48) that the computation of gradients with respect to the hyperparameters is significantly more time consuming than the evaluation of gradients with respect to the latent values.

In the figure 7.6, there are shown the times needed for drawing one sample from joint posterior $p(\theta, \mathbf{f}\,|D)$, and from the conditional posteriors $p(\theta|\,\mathbf{f}, D)$ and $p(\mathbf{f}\,|\theta, D)$, as a function of the number of inducing inputs with FITC approximation and 915 data points. The times are obtained by using 5 leapfrog steps in sampling from $p(\theta|\,\mathbf{f}, D)$ and 10 leapfrog steps in sampling from $p(\mathbf{f}\,|\theta, D)$. These options were used with cerebral vascular diseases and represent the time needed to draw one non-efficient sample with Intel Pentium 4 (1700MHz, 1GB memory) workstation. In the case of alcohol related diseases, the number of leapfrog steps was same but when sampling from $p(\theta|\,\mathbf{f}, D)$ there were 3 trajectories taken at each round. Thus the time needed for one non-efficient sample increased to 24 seconds.

From tables 7.1 and 7.2 it can be summarized that with Intel Pentium 4 (1700MHz, 1GB memory) work station and 915 data points, it took approximately 6.5 hours at minimum (cerebral vascular diseases, FITC approximation and exponential covariance function). With 3193 datapoints the respective time was 37 hours (cerebral vascular diseases, FITC approximation and exponential covariance function).

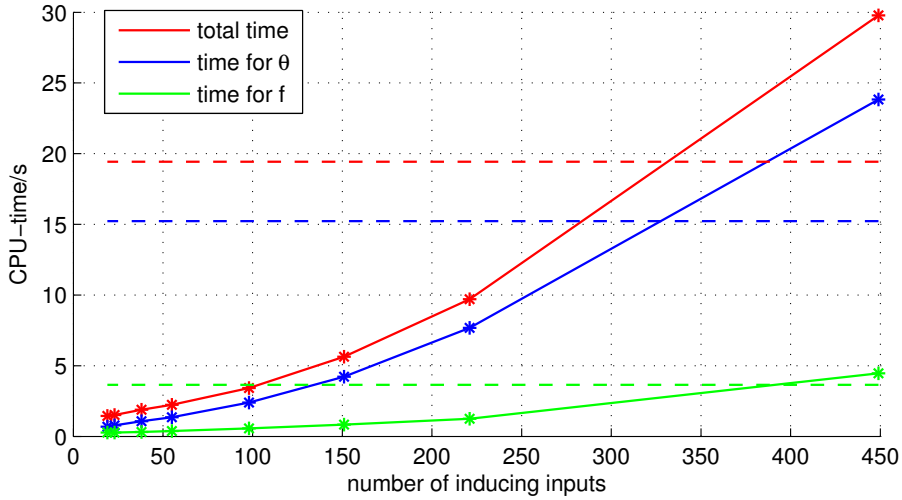


Figure 7.6: **CPU-time for one sample from $p(\theta, \mathbf{f}|D)$ as a function of a number of inducing inputs with 915 data points and FITC approximation**. The continues lines represent the times needed for one sample with FITC approximation and the dashed lines show the time with full GP. From the picture it is seen that the sampling time with FITC is less than the time with full GP when the number of inducing inputs is approximately 40% of the number of data points. In order to compare the model performance see tables 7.3 and 7.4. The times are obtained by using 5 leapfrog steps in sampling from $p(\theta|\,\mathbf{f},\, D)$ and 10 leapfrog steps in sampling from $p(\mathbf{f}\,|\theta,\, D)$ with Intel Pentium 4 (1700MHz, 1GB memory) workstation.

7.4 Model comparison

The model comparison is conducted by the deviance information criterion described in the section 3.1.4. The posterior distributions of covariance function parameters were also compared to each others in order to recognize possible major faults. The distributions are shown in the figure 7.7 and 7.8.

In the case of cerebral vascular diseases, the posterior distribution of length-scale l of a given covariance function and disease is rather similar no matter if the model is full GP or FITC sparse approximation or, if the data was in $20km \times 20km$ or $10km \times 10km$ lattice. However the posterior mean of the magnitude σ^2 in the case of $10km \times 10km$ lattice data is larger than in the case of $20km \times 20km$ lattice. The higher values in the magnitude suggest that there is a need for larger variance with the $10km \times 10km$ lattice. This may be a result from a phenomenon with small length-scale which was not present

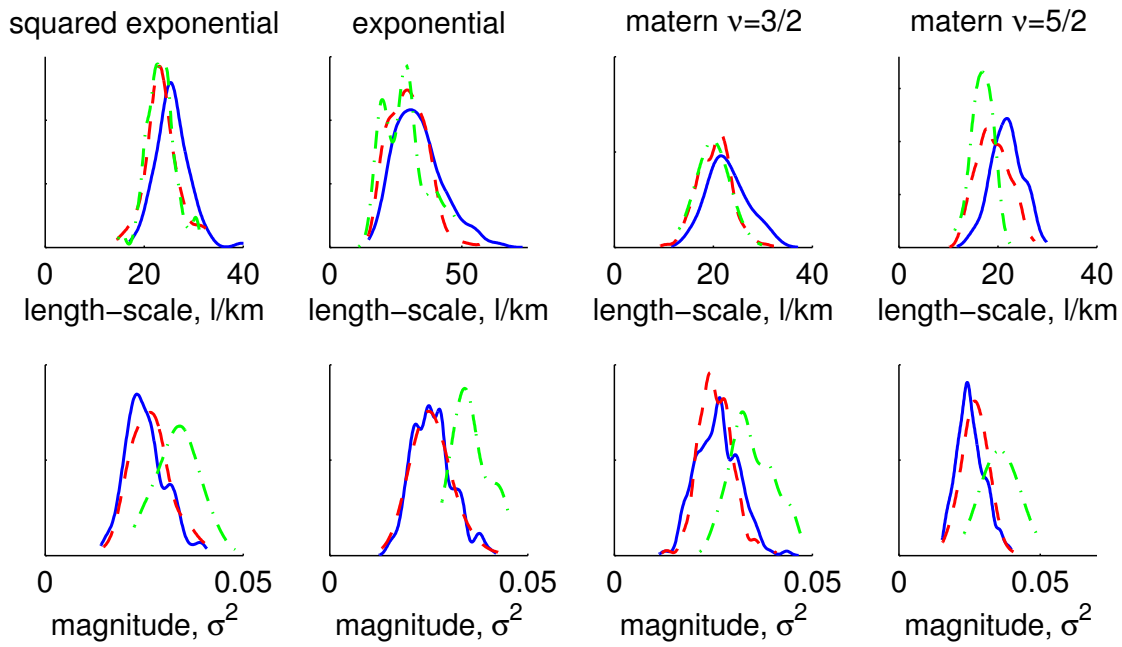


Figure 7.7: **Posterior distributions of length scale and magnitude in cerebral vascular diseases data**. Blue solid line represents FITC approximation and 20km × 20km lattice, red dashed line (- -) represents full GP and 20km × 20km lattice and green dashed line (-.) represents FITC approximation and 10km × 10km lattice.

in the less accurate data of $20km \times 20km$ lattice, but could be seen in the data aggregated into a denser grid. In that case the model may have fitted into the overall phenomenon with longer length scale and since the long length scale does not allow quick variations the model fit to them by increased variance. In the case of alcohol related diseases the length-scale is little smaller in the case of 10km×10km lattice, which suggests as well that there is present a phenomenon with smaller length-scale.

The DIC measure and the number of effective parameters p_D are evaluated with the saturated deviance (Spiegelhalter et al., 2002)

$$D_{\text{st}}(y,\theta) = 2\Sigma_i \left[Y_i \log\left(\frac{Y_i}{E_i\mu_i}\right) - (Y_i - E_i\mu_i) \right]. \tag{7.1}$$

The model complexity, or the number of effective parameters p_D, is not invariant on the choice of the parameterization of $\hat{\theta}$. Although, normally the choice of the parameterization does not have strong effect on it. However, Spiegelhalter et al. (2002) have used

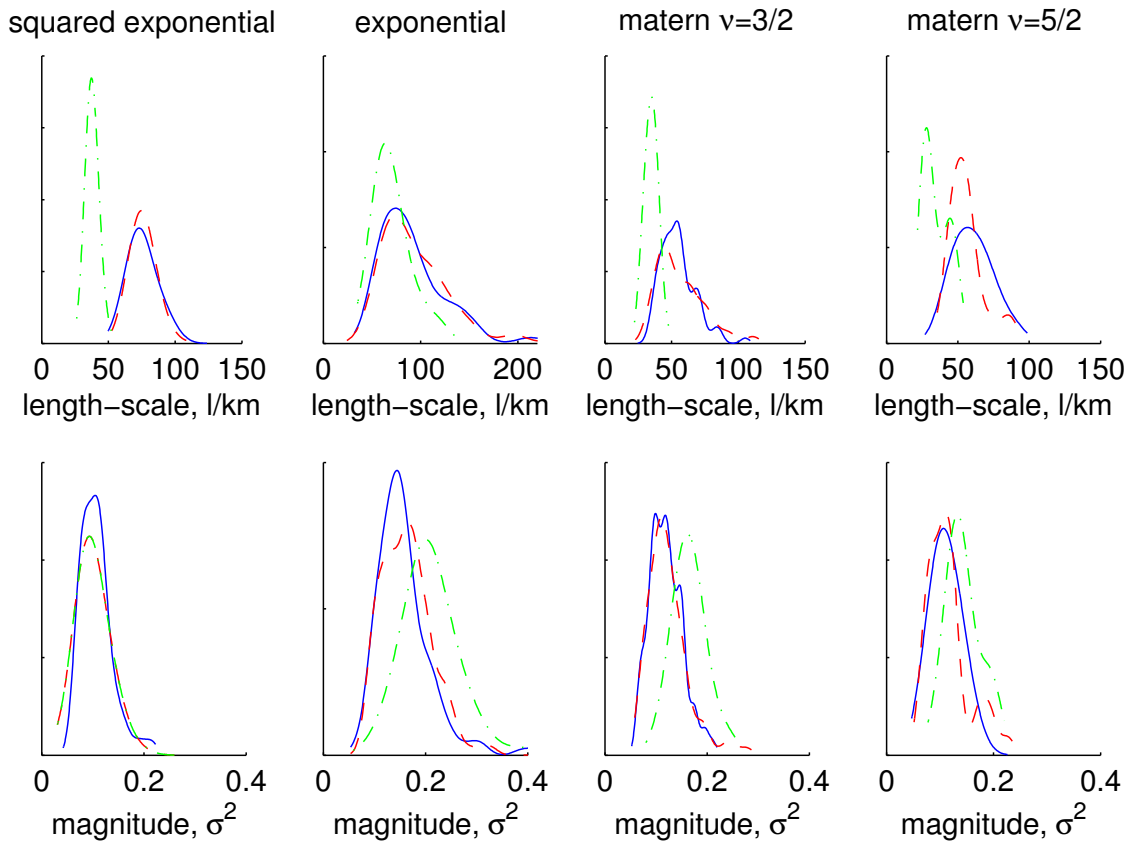


Figure 7.8: **Posterior distributions of length scale and magnitude in cerebral vascular diseases data**. Blue solid line represents FITC approximation and 20km × 20km lattice, red dashed line (- -) represents full GP and 20km × 20km lattice and green dashed line (-.) represents FITC approximation and 10km × 10km lattice.

both mean and median parameterization for the Poisson likelihood, and pointed out that in the case of possible big difference between the deviance and p_D estimates with different characterizations the model should be investigated more carefully. Thus, here the summaries are evaluated with both $\hat{\theta}_{\text{mean}}$ and $\hat{\theta}_{\text{median}}$. The results of the statistics are shown in the tables 7.3–7.6, and it can be concluded that the results with either one of the parameterization are very similar.

From tables 7.3 and 7.4 it can be concluded that in general the full GP models did somewhat better than the sparse models. However, the difference was negligible in most of the cases, the squared exponential and Mátern $\nu = 5/2$ in table 7.3 and the Mátern $3/2$ in table 7.4 being the exceptions, and thus the difference in model performance between full

Table 7.3: **The DIC statistics in the case of cerebral vascular diseases data and 20km × 20km lattice using two alternative parameterization (mean and median) and saturated deviance**. The number of data points in the models is 915 and the number of inducing inputs 221 (approximately 25% of the data points).

	full/ FITC	$\widehat{D}_{\text{avg,st}}(y)$	p_D^{mean}	$\text{DIC}_{\text{st}}^{\text{mean}}$	p_D^{median}	$\text{DIC}_{\text{st}}^{\text{median}}$
k_{sexp}	FITC	904	148.6	1052	143.3	1047
	full	902	144.2	1046	137.7	1040
k_{exp}	FITC	905	142.7	1047	138.6	1043
	full	900	145.2	1045	141.2	1041
$k_{\nu=3/2}$	FITC	905	146.3	1051	141.4	1046
	full	900	147.6	1047	143.8	1043
$k_{\nu=5/2}$	FITC	907	142.7	1050	139.6	1047
	full	894	150.1	1044	147.3	1041

Table 7.4: **The DIC statistics in the case of alcohol related diseases data and 20km × 20km lattice using two alternative parameterization (mean and median) and saturated deviance**. The number of data points in the models is 915 and the number of inducing inputs 221 (approximately 25% of the data points).

	full/ FITC	$\widehat{D}_{\text{avg,st}}(y)$	p_D^{mean}	$\text{DIC}_{\text{st}}^{\text{mean}}$	p_D^{median}	$\text{DIC}_{\text{st}}^{\text{median}}$
k_{sexp}	FITC	1020	52.6	1073	52.7	1073
	full	1020	52.2	1072	52.9	1073
k_{exp}	FITC	902	115.3	1017	117.5	1019
	full	909	112.2	1021	112.8	1022
$k_{\nu=3/2}$	FITC	953	88.8	1042	87.6	1041
	full	954	84.9	1039	81.9	1036
$k_{\nu=5/2}$	FITC	981	75.4	1057	71.4	1053
	full	994	63.4	1057	62.9	1057

and FITC approximation, with the chosen inducing inputs, can be considered to be negligible and perhaps using even less inducing inputs would be enough. The future studies could thus consider also the number of needed inducing inputs.

In the case of cerebral vascular diseases and the 20km×20km lattice the differences in model performance with different covariance functions were really small and all the models did practically as well. However, with the 10km×10km lattice data the exponential covariance function seems to work somewhat better than the others.

Table 7.5: **The DIC statistics in the case of cerebral vascular diseases and 10km×10km lattice using two alternative parameterization (mean and median) and saturated deviance**.

	$\widehat{D}_{\text{avg,st}}(y)$	p_D^{mean}	$\text{DIC}_{\text{st}}^{\text{mean}}$	p_D^{median}	$\text{DIC}_{\text{st}}^{\text{median}}$
k_{sexp}	2936	216.3	3153	213.2	3149
k_{exp}	2891	243.4	3134	241.4	3132
$k_{\nu=3/2}$	2914	231.3	3145	228.7	3143
$k_{\nu=5/2}$	2908	237.4	3145	236.6	3144

Table 7.6: **The DIC statistics in the case of alcohol related diseases and 10km×10km lattice using two alternative parameterization (mean and median) and saturated deviance**.

	$\widehat{D}_{\text{avg,st}}(y)$	p_D^{mean}	$\text{DIC}_{\text{st}}^{\text{mean}}$	p_D^{median}	$\text{DIC}_{\text{st}}^{\text{median}}$
k_{sexp}	2252	195.0	2447	208.3	2461
k_{exp}	2243	176.7	2419	178.1	2421
$k_{\nu=3/2}$	2272	170.5	2443	173.4	2446
$k_{\nu=5/2}$	2331	137.0	2468	131.7	2463

With the alcohol related diseases data set the exponential covariance function worked best with both aggregation level. The number of effective parameter of exponential covariance function model is around 115 (depending on the parameterization and GP) in table 7.4, whereas the number is reduced to almost half in the case of squared exponential and the Mátern $\nu = 5/2$ covariance functions and to 3/4th in the case of Mátern $\nu = 3/2$. This indicates that the model with exponential covariance function is more complex, and it has fitted to data more than the others. However, the difference in DIC value is rather large in favor of exponential function and thus it can be considered working best.

Chapter 8

Conclusions and future work

The aim of this work was to study a hierarchical three level model in disease mapping with a given point referenced healthcare data. The particular model constructed of Poisson likelihood and sparse Gaussian process prior. The sparse approximation used was fully independent training conditional, and the main emphasis of the work was placed on implementing it for the Poisson likelihood. The models were constructed under Bayesian framework and the posterior inference was performed using Markov Chain Monte Carlo methods.

The posterior simulations of latent values were sped up with a transformation using their approximate posterior precision. The transformation worked well and enabled good mixing in the latent value sampling. The efficiency of the posterior simulations was limited by the sampling of covariance function parameters, which seemed to be highly data dependent.

The hierarchical model was constructed with both full and sparse Gaussian process with four different covariance functions, squared exponential, exponential, Mátern $\nu = 3/2$ and Mátern $\nu = 5/2$. The resulting eight models were compared to each other using two sets of mortality data. The model comparison was performed using deviance information criterion. In both of the data cases, the models with exponential covariance function were found best, and the difference between the other models small.

The work was focused on the methodology research and thus the significance of the results for the research of spatial epidemiology in Finland remains still for further study. This will be performed in collaboration with healthcare specialists.

Here the sparse approximation for Gaussian process was compared with the computationally more demanding full Gaussian process. The results were promising and thus encourage for further study of the method. However, probably the most widely used family of models in disease mapping is the conditional autoregressive models, which outperform Gaussian processes in their computation speed but face considerable problems, for example, with areally sparse data. Later the results with the sparse and full Gaussian process will be compared to the results obtained with a conditional autoregressive model. The Gaussian processes used in this work have only one covariance function, which can fit only in a phenomenon with one length-scale. In the future, models with more than one covariance function will be tried in order to test if there are phenomenon with different length-scales. The performance of the models with less inducing inputs would also be of interest.

A technical subject that needs more development is the sampling of the number and the locations of inducing inputs. The accuracy of variational type approximations, such as expectation propagation algorithm, for marginalizing over latent values could also be of interest in future development. At the moment the limitation of the method is the Markov chain Monte Carlo simulations of covariance function parameters, especially the length-scale, and thus the implementation of method in larger data sets needs development in the hyperparameter sampling.

Gaussian processes can be used in wide variety of supervised learning problems. In the thesis the fully independent training conditional approximation was implemented in general manner for Matlab so that it can be used in other problems also. In the future the sparse Gaussian process will be studied and compared to MLP network in a metal casting problem.

At the moment the software is implemented in Matlab environment, which provides an efficient user interface and a wide variety of ready made toolboxes. However Matlab is

not computationally most efficient in large matrix evaluations with known matrix properties. In the future the most time consuming matrix evaluations will be implemented with C/C++ code as mex-files that can be called from Matlab in order to obtain time savings.

Bibliography

Abrahamsen, P. (1997). A review of Gaussian random fields and correlation functions, second edition. Technical Report 917, Norwegian Computing Center.

Abramowitz, M. and Stegun, I. A. (1970). *Handbook of mathematical functions*. Dover Publications, Inc.

Ahmad, O. B., Boschi-Pinto, C., Lopez, A. D., Murray, C. J., Lozano, R., and Inoue, M. (2000). Age standardization of rates: A new WHO standard. *GPE Discussion Paper Series*, 31.

Anderson, R. N. and Rosenberg, H. M. (1998). Age standardization of death rates: Implementation of the year 2000 standard. *National Vital Statistics Reports*, 47(3):1–17.

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2004). *Hierarchical Modelling and Analysis for Spatial Data*. Chapman Hall/CRC.

Beneš, V., Bodlák, K., Møller, J., and Waagepetersen, R. (2002). Bayesian analysis of log Gaussian Cox processes for disease mapping. Technical report, Department of Mathematical Sciences, Aalborg University.

Best, N., Richardson, S., and Thomson, A. (2005). A comparison of Bayesian spatial models for disease mapping. *Statistical Methods in Medical Research*, 14:35–59.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

Brooks, S. P. and Gelman, A. (1998). General Methods for Monitoring Convergence of Iterative Simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455.

Christensen, O. F., Roberts, G. O., and Sköld, M. (2006). Robust Markov chain Monte Carlo methods for spatial generalised linear mixed models. *Journal of Computational and Graphical Statistics*, 15:1–17.

Csató, L. and Opper, M. (2002). Sparse online Gaussian processes. *Neural Computation*, 14(3):641–669.

Diggle, P. (2001). Overview of statistical methods for disease mapping and its relationship to cluster detection. In Elliot, P., Wakefield, J., Best, N., and Briggs, D., editors, *Spatial Epidemiology Methods and Applications*. Oxford University Press.

Duane, S., Kennedy, A., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222.

Elliot, P., Wakefield, J., Best, N., and Briggs, D., editors (2001). *Spatial Epidemiology Methods and Applications*. Oxford University Press.

Gelman, A. (2006). Prior distribution for variance parameters in hierarchical models. *Bayesian Analysis*, 1(3):515–533.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. Capman & Hall/CRC, second edition.

Geyer, C. J. (1992). Practical Markov chain Monte Carlo. *Statistical Science*, 7(4):473–511.

Gilks, W., Richardson, S., and Spiegelhalter, D. (1996). Introducing markov chain monte carlo. In Gilks, W., Richardson, S., and Spiegelhalter, D., editors, *Markov Chain Monte Carlo in Practice*. Chapman & Hall.

Golub, G. H. and van Loan, C. F. (1996). *Matrix Computations*. The Johns Hopkins University Press, third edition.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732.

Harville, D. A. (1997). *Matrix Algebra From a Statistician's Perspective*. Springer-Verlag.

Karvonen, M., Moltchanova, E., Viik-Kajander, M., Moltchanov, V., Rytkönen, M., Kuosa, A., and Tuomilehto, J. (2002). Regional inequality in the risk of acute myocardial infarction in finland: A case study of 35- to 74 year-old men. *Heart Drug*, 2:51–60.

Lampinen, J. and Vehtari, A. (2001). Bayesian Approach for Neural Networks – Review and Case Studies. *Neural Networks*, 14(3):7–24.

Lawson, A. B. (2001). *Statistical Methods in Spatial Epidemology*. John Wiley & Sons, Ltd.

MacEachren, A. M., Brewer, C. A., and Pickle, L. W. (1998). Visualizing georeferenced data: representing reliability of health statistics. *Environment and Planning A*, 30:1547–1561.

Minka, T. (2001). *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology.

Møller, J., Syversveen, A. R., and Waagepetersen, R. P. (1998). Log Gaussian Cox processes. *Scandinavian Journal of Statistics*, 25:451–482.

Møller, J. and Waagepetersen, R. P. (2003). *Statistical Inference and Simulation for Spatial Point Processes*. Chapman Hall/CRC.

Moltchanova, E. (2005). *Application of Bayesian Spatial Methods in Health and Population Studies Using Registry Data*. PhD thesis, University of Jyväskylä, Jyväskylä, Finland.

Monmonier, M. (2004). Lying with maps. *Statistical Science*, 20(3):215–222.

Nabney, I. T. (2001). *NETLAB: Algorithms for Pattern Recognition*. Springer.

Neal, R. M. (1993). Probabilistic Inference Using Markov Chain Monte Carlo Methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer.

Quinonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal Of Machine Learning Research*, 6(3):1939–1959.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

Richardson, S., Thomson, A., Best, N., and Elliot, P. (2004). Interpreting posterior relative risk estimates in disease-mapping studies. *Environmental Health Perspectives*, 112(9):1016–1025.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer, second edition.

Roberts, G. O. (1996). Markov chain concepts related to sampling algorithms. In Gilks, W., Richardson, S., and Spiegelhalter, D., editors, *Markov Chain Monte Carlo in Practice*. Chapman & Hall.

Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16:351–367.

Rytkönen, M. (2004). Not all maps are equal: GIS and spatial analysis in epidemiology. *International Journal of Circumpolar Health*, 64(1).

Seeger, M., Williams, C. K. I., and Lawrence, N. (2003). Fast forward selection to speed up sparse Gaussian process regression. In Bishop, C. M. and Frey, B. J., editors, *Ninth International Workshop on Artificial Intelligence and Statistics*. Society for Artificial Intelligence and Statistics.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of Royal Statistical Society*, 47(1):1–52.

Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian process using pseudo-inputs. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems*, volume 18. The MIT Press.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and van der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of Royal Statistical Society B*, 64(4):583–639.

Staines, A. and Järup, L. (2001). Health event data. In Elliot, P., Wakefield, J., Best, N., and Briggs, D., editors, *Spatial Epidemiology Methods and Applications*. Oxford University Press.

Wahba, G., nad Fangyu Gao, X. L., Xiang, D., Klein, R., and Klein, B. (1999). The bias-variance tradeoff and the randomized GACV. In Kern, M. S., Solla, S. A., and Cohn, D. A., editors, *Advances in Neural Information Processing Systems*, volume 18. The MIT Press.

Walter, S. (2001). Disease mapping: a historical perspective. In Elliot, P., Wakefield, J., Best, N., and Briggs, D., editors, *Spatial Epidemiology Methods and Applications*. Oxford University Press.

Williams, C. K. I. and Barber, D. (1998). Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351.

Williams, C. K. I. and Seeger, M. (2001). Using the Nyström method to speed up kernel machines. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13. The MIT Press.